DESIGNING OF NOVEL INHIBITOR FOR TYPE 2 DIABETES: A MACHINE LEARNING APPROACH

Major project report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

B.GAYATHRI (20UECS0150) **(VTU 16893)**

Under the guidance of
Mr. V. KARTHIKEYAN, M.E.,
ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

DESIGNING OF NOVEL INHIBITOR FOR TYPE 2 DIABETES: A MACHINE LEARNING APPROACH

Major project report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

B. GAYATHRI (20UECS0150) (16893)

Under the guidance of
Mr. V. KARTHIKEYAN, M.E.,
ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

CERTIFICATE

It is certified that the work contained in the project report titled “DESIGNING OF NOVEL INHIBITOR FOR TYPE 2 DIABETES: A MACHINE LEARNING APPROACH” by“B.GAYATHRI (20UECS0150)” has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
May, 2024

Signature of Professor In-charge
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
May, 2024

DECLARATION

I declare that this written submission represented in my own words and where other words have been included, I have adequately cited and referenced the original sources. I also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any source in our submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

B. GAYATHRI

Date: / /

APPROVAL SHEET

This project report entitled DESIGNING OF NOVEL INHIBITOR FOR TYPE 2 DIABETES: A MACHINE LEARNING APPROACH by B.GAYATHRI (20UECS0150) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners	**Supervisor**
	Mr. V. KARTHIKEYAN, M.E.,

Date: / /

Place:

ACKNOWLEDGEMENT

I express our deepest gratitude to our respected President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S. Chairperson Managing Trustee and Vice President.

I am very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN,** for providing us with an environment to complete our project successfully.

I record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. V. SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

I am thankful to our **Head, Department of Computer Science & Engineering,**
Dr. M. S. MURALI DHAR, M.E., Ph.D., for providing immense support in all our endeavors.

I also take this opportunity to express a deep sense of gratitude to our **Internal Supervisor**
Mr. V. KARTHIKEYAN, M.E., for his cordial support, valuable information and guidance, he helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M.Tech., Ms. C. SHYAMALA KUMARI, M.E.,** for their valuable guidance and support throughout the course of the project.

I thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

B. GAYATHRI (20UECS0150)

ABSTRACT

Type 2 diabetes mellitus (T2DM) is a chronic metabolic disorder affecting millions worldwide, characterized by insulin resistance and impaired insulin secretion. Despite the availability of various anti-diabetic medications, there is a growing need for more effective and safer therapeutics. In this, i propose a novel approach utilizing machine learning algorithms for the design of inhibitors targeting key molecular pathways involved in T2DM pathogenesis.Random Forest is a versatile and widely used machine learning algorithm, particularly effective for classification and regression tasks. It belongs to the ensemble learning methods, which involve combining multiple models to improve prediction accuracy. At its core, Random Forest consists of a collection of decision trees. Each decision tree is built using a subset of the training data and a random subset of the features. Decision trees are simple models that recursively split the data based on the features to make predictions.Random Forest is known for its good performance across a wide range of datasets and often serves as a baseline model for many classification and regression tasks. However, achieving optimal accuracy requires careful tuning of hyperparameters and preprocessing of the data. Additionally, it's essential to assess the model's performance on unseen data to avoid overfitting. Overall accuracy rate of 95%.

Keywords: Type 2 diabetes mellitus, inhibitor design, random forest, drug discovery.

LIST OF FIGURES

4.1 **General Architecture** 11
4.2 **Data Flow Diagram** 12
4.3 **Use Case Diagram** 13
4.4 **Class Diagram** 14
4.5 **Sequence Diagram** 15
4.6 **Collaboration Diagram** 16
4.7 **Activity Diagram** 17

5.1 **Molecular Structure** 23
5.2 **Model Prediction** 24
5.3 **Integration Testing** 26
5.4 **Integration Testing** 28
5.5 **System Testing** 30
5.6 **Test Result** 31

6.1 **Descriptors Calculation** 35
6.2 **Data Frames** 36

8.1 **Offer Letter** 39

9.1 **Plagiarism** 41

10.1 **Poster** 49

LIST OF ACRONYMS AND ABBREVIATIONS

AI	Artificial Intelligences
CV	Computer Vision
DD	Drug Discovery
DT	Decision Tree
MD	Molecular Docking
ML	Machine Learning
RF	Random Forest
SVM	Support Vector Machine
T2DM	Type 2 Diabetes Mellitus

TABLE OF CONTENTS

Page.No

ABSTRACT v

LIST OF FIGURES vi

LIST OF ACRONYMS AND ABBREVIATIONS vii

1 INTRODUCTION 1
- 1.1 Introduction 1
- 1.2 Aim of the Project 2
- 1.3 Project Domain 2
- 1.4 Scope of the Project 2

2 LITERATURE REVIEW 3

3 PROJECT DESCRIPTION 6
- 3.1 Existing System 6
- 3.2 Proposed System 6
- 3.3 Feasibility Study 7
  - 3.3.1 Economic Feasibility 8
  - 3.3.2 Technical Feasibility 8
  - 3.3.3 Social Feasibility 8
- 3.4 System Specification 9
  - 3.4.1 Hardware Specification 9
  - 3.4.2 Software Specification 10
  - 3.4.3 Standards and Policies 10

4 METHODOLOGY 11
- 4.1 General Architecture 11
- 4.2 Design Phase 12
  - 4.2.1 Data Flow Diagram 12
  - 4.2.2 Use Case Diagram 13
  - 4.2.3 Class Diagram 14

4.2.4 Sequence Diagram 15
4.2.5 Collaboration Diagram 16
4.2.6 Activity Diagram 17
4.3 Algorithm & Pseudo Code 18
4.3.1 Algorithm 18
4.3.2 Pseudo Code 18
4.4 Module Description 19
4.4.1 Data Collection 19
4.4.2 Exploratory Data Analysis 19
4.4.3 Descriptor Calculation 20
4.5 Steps to execute/run/implement the project 20
4.5.1 Model Building 20
4.5.2 Model Comparison 21
4.5.3 Model Development 21

5 IMPLEMENTATION AND TESTING 23
5.1 Input and Output 23
5.1.1 Input Design 23
5.1.2 Output Design 24
5.2 Testing 25
5.3 Types of Testing 25
5.3.1 Unit Testing 25
5.3.2 Integration Testing 27
5.3.3 System Testing 29
5.3.4 Test Result 31

6 RESULTS AND DISCUSSIONS 32
6.1 Efficiency of the Proposed System 32
6.2 Comparison of Existing and Proposed System 32
6.3 Sample Code 34

7 CONCLUSION AND FUTURE ENHANCEMENTS 37
7.1 Conclusion 37
7.2 Future Enhancements 37

8 INDUSTRY DETAILS **38**

8.1 Industry name:Qstatix private limited 38

8.1.1 Duration of Internship (FEB 10 - MAY 30) 38

8.1.2 Duration of Internship in months: 4 MONTHS 38

8.1.3 Industry Address: PLOT NO :5, Mohan nagar chowrasta Nagole to kothapet road, Nagole Hyderabad-500035 38

8.2 Internship Offer Letter 38

8.3 Internship Completion Certificate 40

9 PLAGIARISM REPORT **41**

10 SOURCE CODE & POSTER PRESENTATION **42**

10.1 Source Code 42

10.2 Poster Presentation 49

References **49**

Chapter 1

INTRODUCTION

1.1 Introduction

Type 2 diabetes mellitus (T2DM) represents a significant global health challenge, with its prevalence steadily increasing due to factors such as sedentary lifestyles, unhealthy dietary habits, and aging populations. T2DM is characterized by insulin resistance and impaired insulin secretion, leading to hyperglycemia and various complications, including cardiovascular disease, neuropathy, and nephropathy. Despite the availability of numerous anti-diabetic medications, including insulin sensitizers, insulin secretagogues, and incretin-based therapies, many patients fail to achieve adequate glycemic control or suffer from adverse effects associated with current treatments.

Moreover, existing drugs primarily focus on managing symptoms rather than addressing the underlying molecular mechanisms driving T2DM pathogenesis.In recent years, there has been a growing interest in employing computational methods, particularly ML and AI, to accelerate drug discovery and development processes. ML algorithms offer the potential to analyze large datasets, identify patterns, and predict the biological activity of compounds against specific molecular targets, thereby facilitating the design of novel therapeutics with improved efficacy and safety profiles.

In this study, i proposed a novel approach to designing inhibitors for T2DM using a machine learning framework. The nature of the problem (classification or regression), the distribution of classes or target values, and the presence of imbalanced data can all influence the accuracy of the Random Forest algorithm. Random Forest randomly selects a subset of features at each node for splitting. Having a larger pool of features can sometimes improve accuracy by increasing the diversity of trees in the forest. The Random Forest algorithm's accuracy depends heavily on the quality and relevance of the training data. High-quality, representative data usually leads to better accuracy.

1.2 Aim of the Project

The aim of this project is to utilize a machine learning approach to design novel inhibitors for the treatment of Type 2 Diabetes Mellitus (T2DM). Specifically, the project seeks to achieve the objectives.Employ advanced machine learning techniques, including deep learning and ensemble learning, to develop predictive models capable of accurately classifying the inhibitory potential of small molecules against T2DM-associated targets.By achieving these objectives, the project aims to expedite the drug discovery process for T2DM, offering a data-driven and computationally guided approach to the design of novel therapeutics with improved efficacy and reduced side effects compared to existing treatments.

1.3 Project Domain

Utilizing advanced machine learning algorithms, such as deep learning and ensemble learning, to analyze large datasets of molecular structures and bioactivity profiles. These techniques enable the development of predictive models for identifying potential inhibitors with therapeutic relevance.By operating at the intersection of these domains, the project aims to leverage computational approaches to accelerate the discovery of novel inhibitors for T2DM, addressing a critical unmet need in the field of diabetes therapeutics.

1.4 Scope of the Project

The scope of the project encompasses several key aspects related to the design of novel inhibitors for Type 2 Diabetes Mellitus (T2DM) using a machine learning approach. Gathering a diverse and comprehensive dataset of molecular structures and their corresponding bioactivity profiles against validated targets implicated in T2DM pathogenesis. This involves sourcing data from various public repositories and literature databases. Employing advanced machine learning techniques, such as Random Forest randomly selects a subset of features at each node for splitting. Having a larger pool of features can sometimes improve accuracy by increasing the diversity of trees in the forest.. This includes data preprocessing, feature engineering, model training, and evaluation.

Chapter 2

LITERATURE REVIEW

[1]Alaa AM, et al.,(2019) Introduced the Missing data on 423,604 participants without CVD at baseline in UK Biobank, we developed a ML-based model for predicting CVD risk based on 473 available variables. Our ML-based model was derived using AutoPrognosis, an algorithmic tool that automatically selects and tunes ensembles of ML modeling pipelines (comprising data imputation, feature processing, classification and calibration algorithms). We compared our model with a well-established risk prediction algorithm based on conventional CVD risk factors (Framingham score), a Cox proportional hazards (PH) model based on familiar risk factors (i.e, age, gender, smoking status, systolic blood pressure, history of diabetes, reception of treatments for hypertension and body mass index), and a Cox PH model based on all of the 473 available variables.

[2]Brnabic A, et al.,(2018).Introduced the Methods of analyzing real world evidence (RWE) generally provide estimates that provide estimates for parameters of a population of patients, whereas the application of RWE to address decision making for the individual patient is less well established. The objective of this study was to review the literature to identify methods used for patient-level decision making using observational, real-world data sources.

[3]Fröhlich H,et al.,(2018)Introduced Personalized, precision, P4, or stratified medicine is understood as a medical approach in which patients are stratified based on their disease subtype, risk, prognosis, or treatment response using specialized diagnostic tests. The key idea is to base medical decisions on individual patient characteristics, including molecular and behavioral biomarkers, rather than on population averages. Personalized medicine is deeply connected to and dependent on data science, specifically machine learning (often named Artificial Intelligence in the mainstream media).

[4]Gawehn E,et al.,(2016) Introduced their first heyday in molecular informatics

and drug discovery approximately two decades ago. Currently, we are witnessing renewed interest in adapting advanced neural network architectures for pharmaceutical research by borrowing from the field of “deep learning”. Compared with some of the other life sciences, their application in drug discovery is still limited.

[5]Grote T, et al., (2020).In recent years, they Introduced a plethora of high- profile scientific publications has been reporting about machine learning algorithms outperforming clinicians in medical diagnosis or treatment recommendations. This has spiked interest in deploying relevant algorithms with the aim of enhancing decision-making in healthcare. In this paper, we argue that instead of straightforwardly enhancing the decision-making capabilities of clinicians and healthcare institutions, deploying machines learning algorithms entails trade-offs at the epistemic and the normative level.

[6]Hische M,et al.(2010). Introduced the prevalence of unknown impaired fasting glucose (IFG), impaired glucose tolerance (IGT), or type 2 diabetes mellitus (T2DM) is high. Numerous studies demonstrated that IFG, IGT, or T2DM are associated with increased cardiovascular risk, therefore an improved identification strategy would be desirable.

[7]Hill NR,et al.,(2019) Introduced to develop and evaluate novel and conventional statistical and machine learning models for risk-predication of AF. This was a retrospective, cohort study of adults (aged 30 years) without a history of AF, listed on the Clinical Practice Research Datalink, from January 2006 to December 2016. Models evaluated included published risk models (Framingham, ARIC, CHARGE-AF), machine learning models, which evaluated baseline and time-updated information (neural network, LASSO, random forests, support vector machines), and Cox regression.The optimal time-varying machine learning model exhibited greater predictive performance.

[8]Kim I, et al.,(2019)Introduced For the high RS group, accuracy was 0.903 through Two-class Decision Jungle method in test set. For the low RS group, the accuracy was 0.726 when the Two-class Neural Network method was applied. The AUC of the ROC curve was 0.917 in the high RS group and 0.744 in the low RS group in test set. In addition, we conducted an internal validation using 76 patients

who underwent ODX testing between January 2017 and July 2017. The accuracy of validation was 0.880 in the high RS group and 0.790 in the low RS group.We developed a predictive model using machine learning that could represent a useful and easy-to-access tool for the selection of high ODX RS patients

[9]Luo W,et al.,(2016)Introduced to attain a set of guidelines on the use of machine learning predictive models within clinical settings to make sure the models are correctly applied and sufficiently reported so that true discoveries can be distinguished from random coincidence.A multidisciplinary panel of machine learning experts, clinicians, and traditional statisticians were interviewed, using an iterative process in accordance with the Delphi method.

[10]Vamathevan J,et al.,(2019) Introduced the Drug discovery and development pipelines are long, complex and depend on numerous factors. Machine learning (ML) approaches provide a set of tools that can improve discovery and decision making for well-specified questions with abundant, high-quality data. Opportunities to apply ML occur in all stages of drug discovery. Examples include target validation, identification of prognostic biomarkers and analysis of digital pathology data in clinical trials.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

There are several existing machine learning algorithms that can be applied to the task of designing novel inhibitors for Type 2 Diabetes Mellitus (T2DM). Decision Trees recursively partition the feature space into regions, with each region corresponding to a simple decision rule.They are easy to interpret and visualize, making them useful for understanding the relationship between features and outcomes.

Decision trees are a fundamental machine learning algorithm used in various domains, including drug discovery, due to their simplicity, interpretability, and effectiveness. In the context of designing novel inhibitors for Type 2 Diabetes Mellitus (T2DM), decision trees can be applied to identify key molecular features and their relationships with inhibitory activity. Overall, decision trees offer a versatile and interpretable approach for designing novel inhibitors for Type 2 Diabetes Mellitus, enabling researchers to uncover meaningful insights from molecular data and guide drug discovery efforts.

Disadvantages:

- Random Forest models are often considered black boxes, making it challenging to understand their decision-making process.
- Despite being less prone to overfitting than individual decision trees, Random Forest models can still overfit to noisy or irrelevant features.
- Balancing bias and variance in Random Forest models can be challenging, particularly in situations with imbalanced or noisy data.

3.2 Proposed System

The proposed system for designing novel inhibitors for Type 2 Diabetes Mellitus (T2DM) integrates ML techniques with computational drug discovery methods to overcome the limitations of traditional approaches. Random Forest is a powerful machine learning algorithm that can be effectively utilized in the task of designing novel inhibitors for Type 2 Diabetes Mellitus (T2DM). Random Forest can identify the most important features (e.g., molecular descriptors) that contribute to inhibitory activity against T2DM. By analyzing the feature importances computed by the model, researchers can gain insights into the molecular properties associated with effective inhibitors.

Although Random Forest is an ensemble method composed of multiple decision trees, it still provides insights into feature importance and model predictions. Researchers can interpret the relative importance of features and understand how they contribute to the prediction of inhibitory activity.By leveraging the capabilities of Random Forest, researchers can build accurate and interpretable models for predicting inhibitory activity against T2DM, facilitating the discovery of novel inhibitors with therapeutic potential.

Advantages:

- The proposed system leverages advanced machine learning algorithms, such as Random Forest or other suitable models, to improve prediction accuracy compared to traditional methods.
- The system utilizes a data-driven approach, leveraging large datasets of molecular descriptors and inhibitory activity against T2DM. By analyzing vast amounts of data, the system can identify patterns and trends that may not be apparent through manual experimentation alone, leading to the discovery of novel inhibitors with improved efficacy.

3.3 Feasibility Study

The feasibility of the project relies on the availability of comprehensive datasets of molecular structures and bioactivity profiles against validated targets implicated in Type 2 Diabetes Mellitus (T2DM) pathogenesis. While public repositories and literature databases provide rich sources of data, ensuring the quality and relevance of the datasets may require careful curation and validation.Developing and validating machine learning models, conducting molecular docking simulations, and performing experimental validation require significant time and resources. Therefore, careful planning and allocation of resources are essential to ensure project milestones are met within the specified timeframe and budget.

3.3.1 Economic Feasibility

One of the primary economic considerations for the proposed system is the cost of computational resources required for developing machine learning models, conducting molecular docking simulations, and analyzing data. This includes expenses associated with hardware infrastructure, such as high-performance computing clusters or cloud computing services, as well as software licenses for ML libraries and molecular docking software. While initial setup costs may be significant, leveraging cloud computing services offers scalability and cost-effectiveness, allowing for flexible utilization based on project requirements. Conducting experimental validation of selected lead compounds through in vitro and/or in vivo assays incurs expenses related to laboratory equipment, consumables, reagents, and personnel salaries. Collaborations with academic or industry partners with existing infrastructure and expertise can help reduce experimental validation costs.

3.3.2 Technical Feasibility

The technical feasibility of the proposed system relies on the availability of comprehensive datasets of molecular structures and their corresponding bioactivity profiles against validated targets implicated in Type 2 Diabetes Mellitus (T2DM) pathogenesis. While publicly available databases and repositories offer rich sources of data, ensuring data quality, relevance, and consistency is essential for robust model development and validation.Advanced machine learning algorithms, such as deep learning and ensemble learning, are essential components of the proposed system for predictive modeling. The technical feasibility of implementing these algorithms depends on the availability of suitable libraries and frameworks, such as TensorFlow, PyTorch, scikit-learn, and XGBoost, as well as expertise in algorithm selection, parameter tuning, and model evaluation.

3.3.3 Social Feasibility

The proposed system for designing novel inhibitors for Type 2 Diabetes Mellitus (T2DM) has significant potential social impact by contributing to the development of more effective and safer treatments for individuals affected by T2DM. By accelerating the discovery of novel therapeutics, the proposed system aims to improve patient outcomes, reduce disease burden, and enhance quality of life for individuals living with T2DM. Ensuring the accessibility and affordability of novel inhibitors developed through the proposed system is essential for maximizing social feasibility. Collaborations with healthcare providers, pharmaceutical companies, and regulatory agencies can help facilitate equitable access to innovative treatments, particularly in underserved communities and low-resource settings where the prevalence of T2DM may be higher.ensuring social feasibility involves adopting a patient-centric, equitable, and ethically responsible approach to drug discovery for T2DM. By addressing social considerations and engaging stakeholders throughout the project lifecycle, the proposed system aims to maximize its positive social impact and contribute to the collective effort to combat T2DM and improve public health outcomes.

3.4 System Specification

- Access and collect comprehensive datasets of molecular structures and bioactivity profiles against validated targets implicated in Type 2 Diabetes Mellitus (T2DM) pathogenesis.
- Curate and preprocess the datasets to ensure data quality, consistency, and relevance for machine learning model development.
- Utilize advanced machine learning techniques, including deep learning and ensemble learning, for predictive modeling of inhibitory activity of small molecules against T2DM-associated targets.
- Use machine learning models to prioritize and identify potential lead candidates with high predicted inhibitory activity against key molecular pathways involved in T2DM.
- Conduct virtual screening of compound libraries to identify promising hits for further optimization.

3.4.1 Hardware Specification

- Processor: Pentium –IV and more
- RAM : 4 GB (min)and moreand more

- Hard Disk: 20 GB and more
- Key Board : Standard Windows Keyboard and more
- Mouse: Two or Three Button Mouse and more
- Monitor: SVGA and more

3.4.2 Software Specification

- Operating System : Windows XP X64 edition
- Coding Language:python
- Front End :python 3.11.9
- Back End:MySQL 9
- IDE : Netbeans 8.1

3.4.3 Standards and Policies

Anaconda Prompt

Anaconda prompt is a type of command line interface which explicitly deals with the ML(MachineLearning) modules.And navigator is available in all the Windows,Linux and MacOS.The anaconda prompt has many number of IDE's which make the coding easier. The UI can also be implemented in python.

Standard Used: ISO/IEC 27001

Jupyter

It's like an open source web application that allows us to share and create the documents which contains the live code, equations, visualizations and narrative text. It can be used for data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning.

Standard Used: ISO/IEC 27001

Chapter 4

METHODOLOGY

4.1 General Architecture

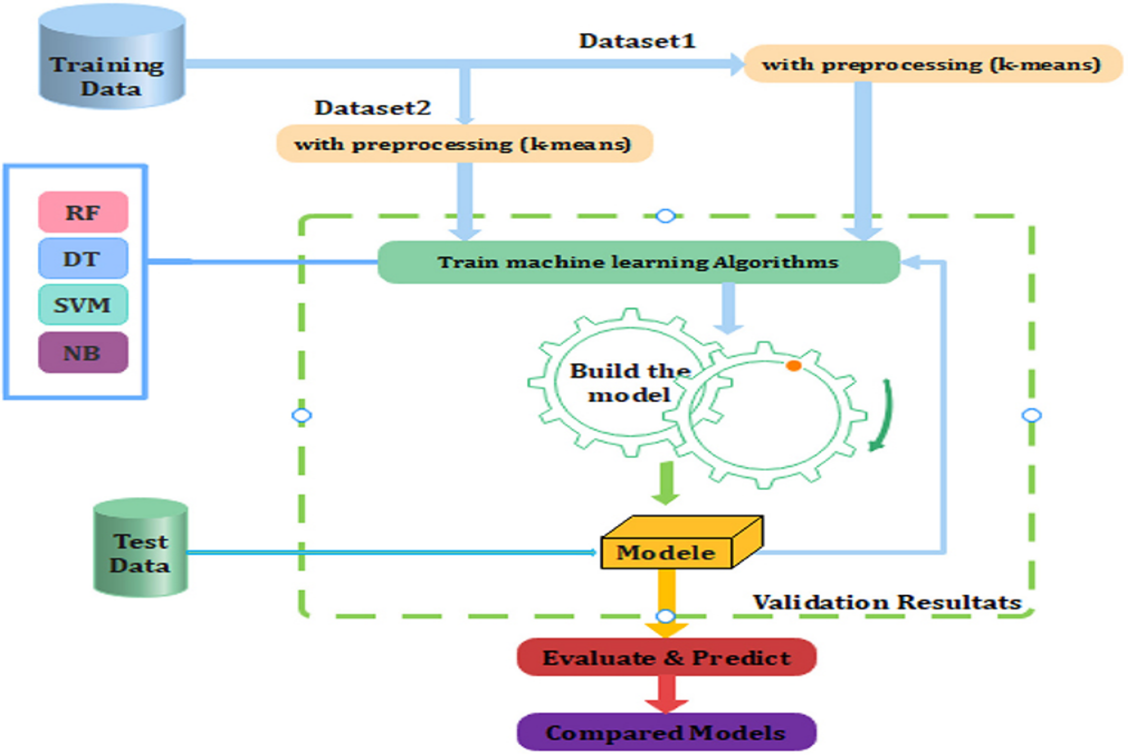


Figure 4.1: **General Architecture**

In this Figure 4.1, the depicted architecture illustrates that Advanced machine learning algorithms, such as random forests, or gradient boosting machines, are utilized to develop predictive models. The models are trained using the curated datasets, with features derived from molecular descriptors, fingerprints, and other relevant properties. Machine learning models are used to prioritize and identify potential lead candidates with high predicted inhibitory activity against key molecular pathways involved in T2DM. Virtual screening techniques may be employed to explore chemical space and identify promising hits for further optimization.

4.2 Design Phase

4.2.1 Data Flow Diagram

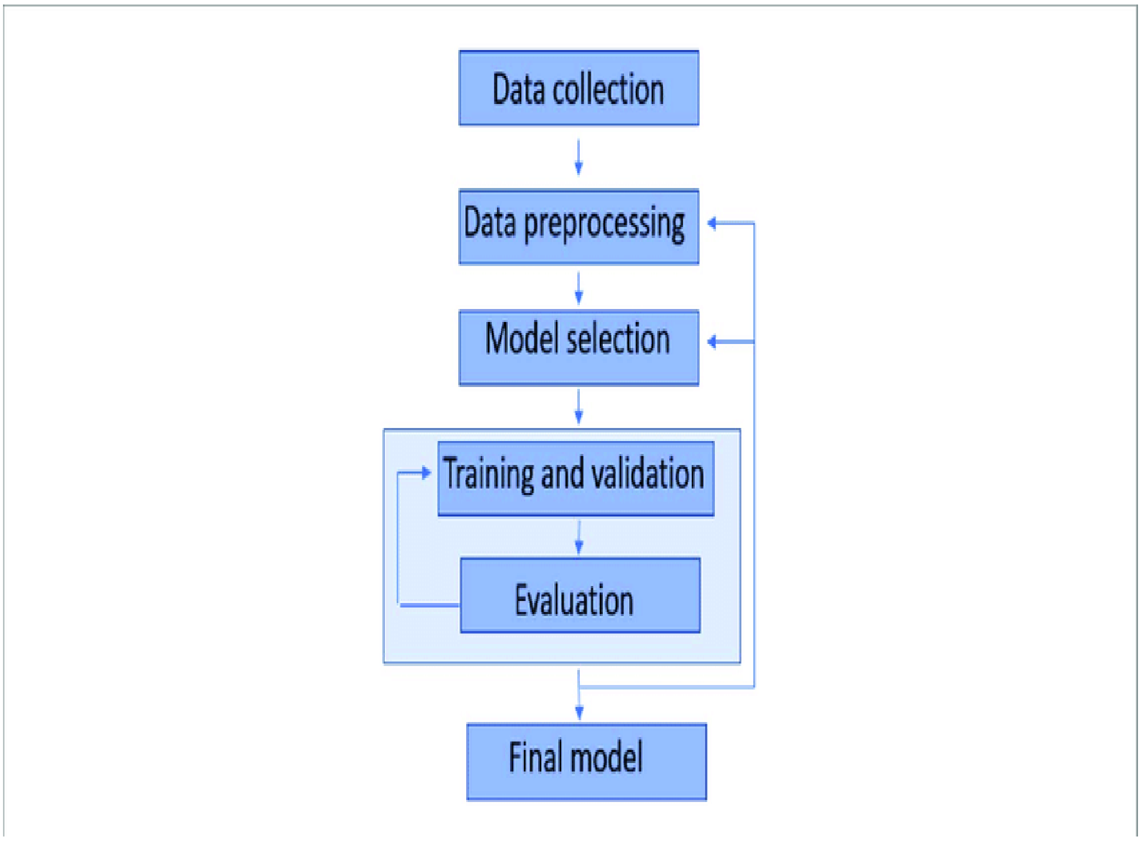


Figure 4.2: **Data Flow Diagram**

In this Figure 4.2 , the data flow illustrates the flow of data and processes within the system architecture for designing a novel inhibitor for Type 2 Diabetes Mellitus (T2DM) using a machine learning approach. Involves the acquisition of comprehensive datasets from external data sources, including data retrieval, parsing, and storage in a central repository. Encompasses preprocessing steps such as data cleaning, normalization, feature extraction, and transformation to prepare the data for model training and analysis. Involves training machine learning models using preprocessed data to predict the inhibitory activity of small molecules against T2DM-associated targets.

4.2.2 Use Case Diagram

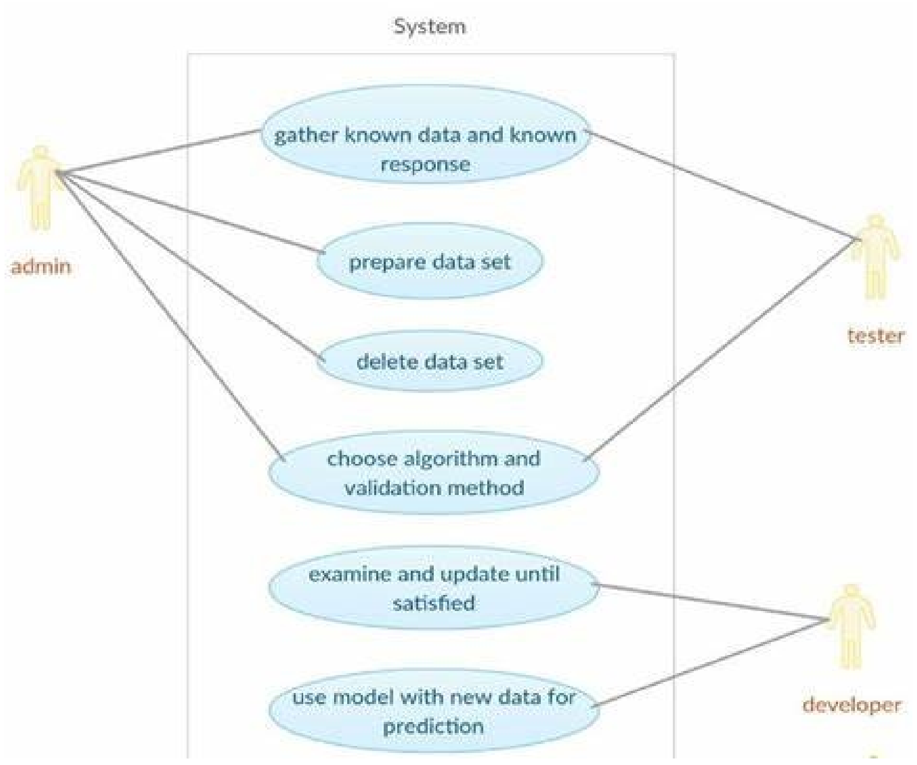

System
gather known data and known response
prepare data set
delete data set
choose algorithm and validation method
examine and update until satisfied
use model with new data for prediction
admin
tester
developer

Figure 4.3: **Use Case Diagram**

In this Figure 4.3 , the use case illustrates the various actors, functionalities, and interactions within the system for designing a novel inhibitor for Type 2 Diabetes Mellitus (T2DM) using a machine learning approach. Allows researchers to acquire comprehensive datasets containing molecular structures and bioactivity profiles related to T2DM from external sources such as public repositories or experimental studies. Enables researchers to preprocess acquired data by cleaning, normalizing, and transforming it to prepare for model training and analysis. Facilitates the training of machine learning models using preprocessed data to predict the inhibitory activity of small molecules against T2DM-associated targets. Allows researchers to conduct molecular docking simulations to validate predicted interactions between designed inhibitors and target proteins implicated in T2DM pathogenesis. Enables researchers to prioritize and identify potential lead compounds with high predicted inhibitory activity based on model predictions and molecular docking results. Allows laboratory personnel to perform experimental validation of selected lead compounds through in vitro and/or in vivo assays to assess biological activity and therapeutic potential.

4.2.3 Class Diagram

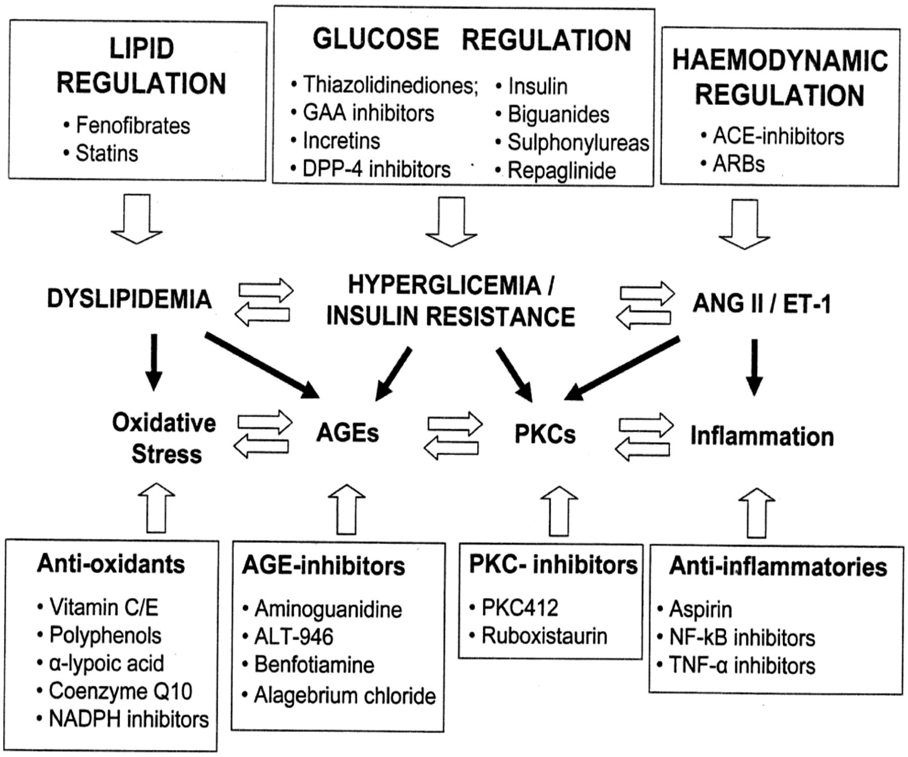


Figure 4.4: **Class Diagram**

In this Figure 4.4 , The class diagram illustrates the static structure of the system for designing a novel inhibitor for Type 2 Diabetes Mellitus (T2DM) using a machine learning approach. It outlines the classes, attributes, and relationships among the objects within the system. Represents the class responsible for acquiring comprehensive datasets containing molecular structures and bioactivity profiles related to T2DM from external sources. Represents the class responsible for preprocessing acquired data by cleaning, normalizing, and transforming it to prepare for model training and analysis. Represents the class responsible for training machine learning models using preprocessed data to predict the inhibitory activity of small molecules against T2DM-associated targets.

4.2.4 Sequence Diagram

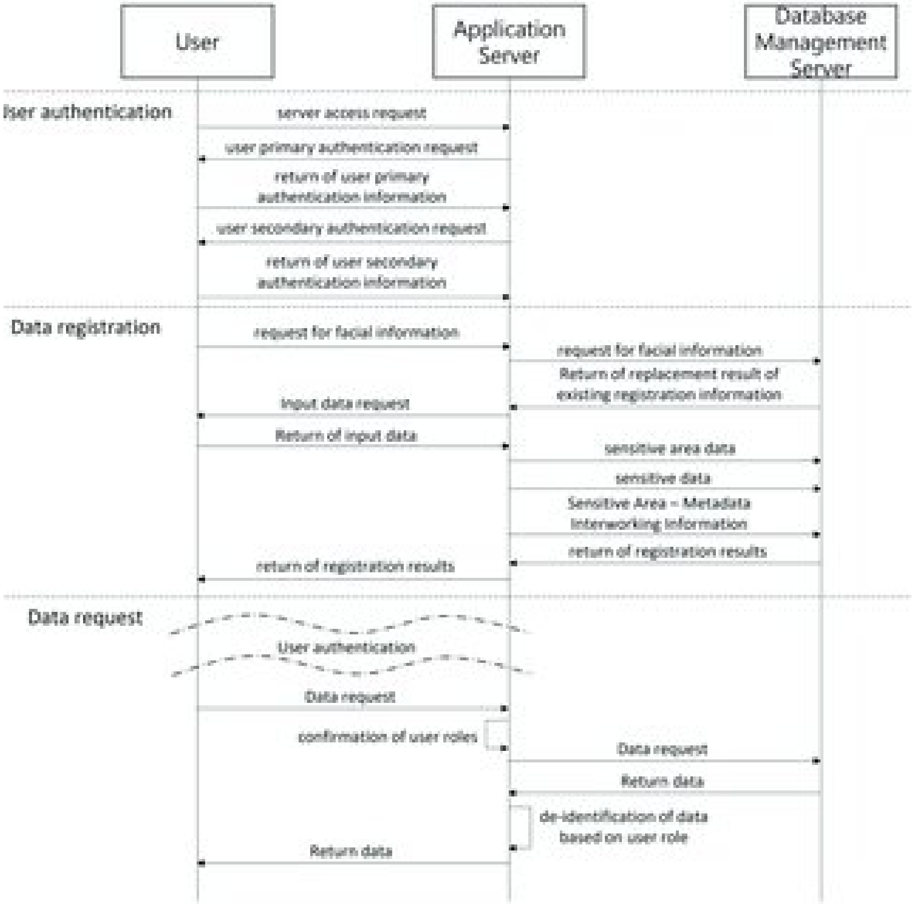

User
Application Server
Database Management Server
Iser authentication
server access request
user primary authentication request
return of user primary authentication information
user secondary authentication request
return of user secondary authentication information
Data registration
request for facial information
request for facial information
Return of replacement result of existing registration information
Input data request
Return of input data
sensitive area data
sensitive data
Sensitive Area – Metadata Interworking Information
return of registration results
return of registration results
Data request
User authentication
Data request
confirmation of user roles
Data request
Return data
de-identification of data based on user role
Return data

Figure 4.5: **Sequence Diagram**

In this Figure 4.5 , The sequence diagram illustrates the interaction between objects or components in a specific scenario or sequence of events within the system for designing a novel inhibitor for Type 2 Diabetes Mellitus (T2DM) using a machine learning approach. This sequence diagram provides a visual representation of the dynamic behavior and interactions within the system architecture, illustrating the sequence of events involved in the process of designing novel inhibitors for Type 2 Diabetes Mellitus using a machine learning approach. The sequence diagram illustrates the sequential order of events within the system, showing how each component interacts with others to perform specific tasks in the process of designing novel inhibitors for T2DM.

4.2.5 Collaboration Diagram

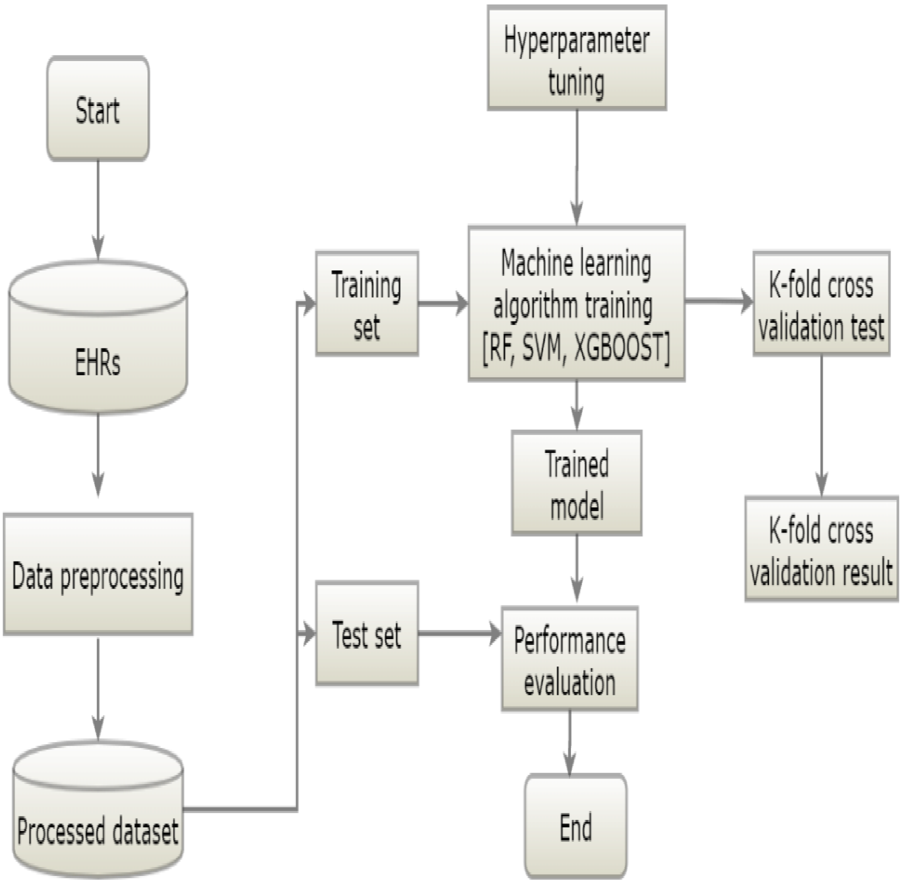


Figure 4.6: **Collaboration Diagram**

In this Figure 4.6 , this diagram illustrates that interactions and relationships among objects or components within the system for designing a novel inhibitor for Type 2 Diabetes Mellitus (T2DM) using a machine learning approach. Collaboration diagrams can also illustrate synchronization and parallelism in the system, showing how multiple objects collaborate concurrently to perform tasks or handle concurrent events.This collaboration diagram offers a clear depiction of the interactions, relationships, and communication among objects or components within the system architecture for designing novel inhibitors for Type 2 Diabetes Mellitus using a machine learning approach.

4.2.6 Activity Diagram

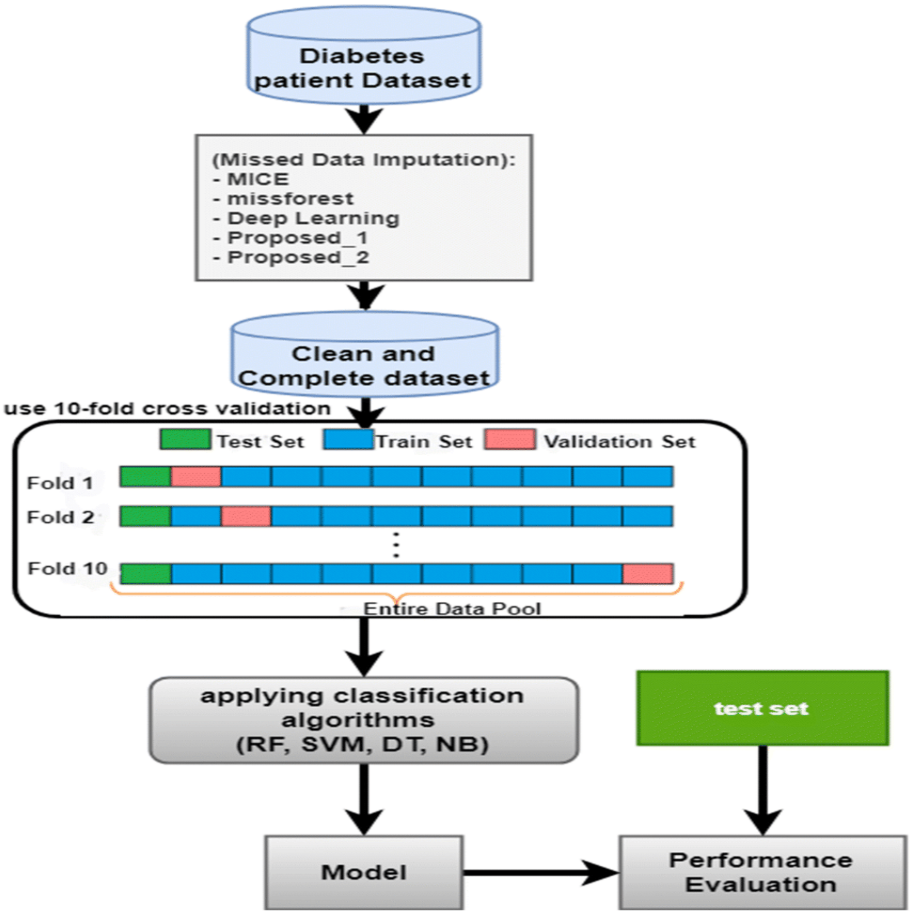


Figure 4.7: **Activity Diagram**

In this Figure 4.7 , an activity diagram illustrates the flow of activities or tasks within the system for designing a novel inhibitor for Type 2 Diabetes Mellitus (T2DM) using a machine learning approach.A horizontal dashed line represents a synchronization bar, indicating synchronization points where concurrent activities wait for all parallel paths to complete before proceeding. Rectangular nodes represent simple actions or tasks performed within the system. Rounded rectangles represent subactivities or complex tasks that may be further decomposed into detailed steps. This activity diagram provides a visual representation of the sequence of activities and tasks involved in the process of designing novel inhibitors for Type 2 Diabetes Mellitus using a machine learning approach, illustrating the flow of control, decision points, concurrency, and synchronization within the system.

4.3 Algorithm & Pseudo Code

4.3.1 Algorithm

Step 1: Start
Step 2: Choose a Dataset
Step 3: Prepare Dataset for Training
Step 4: Create Training Data
Step 5: Shuffle the Dataset
Step 6: Assigning Labels and Features
Step 7: Normalising X and converting labels to categorical data
Step 8: Split X and Y for use in random forest
Step 9: Define, compile and train the random forest Model
Step 10: Accuracy and model building
Step 11: Stop

4.3.2 Pseudo Code

```
    Begin\\
Function EncryptData(data):\\
    encrypted data = Encrypt(data)\\
    return encrypted data\\
Function SearchData(encrypted data, query):\\
    search results = []\\
    for data in encrypted data:\\
        if query in data:\\
            search results.append(data)\\
    return search results\\
Function VerifyData(encrypted data, cryptographic proofs):\\
    for data, proof in zip(encrypted data, cryptographic proofs):\\
        if not Verify(data, proof):\\
            return False\\
    return True\\
Function DecryptData(encrypted data, decryption key):\\
    decrypted data = []\\
    for data in encrypted data:\\
        decrypted data.append(Decrypt(data, decryption key))\\
    return decrypted data\\
Function PerformOperations(encrypted data, operation):\\
    updated data = []\\
    for data in encrypted data:\\
        updated data.append(PerformOperation(data, operation))\\
    return updated data\\
Function HandleErrorsAndSecurity(errors, security concerns):\\
    if errors:\\
        Handle(errors)\\
    if security concerns:\\
        Handle(security concerns)\\
        End
```

4.4 Module Description

4.4.1 Data Collection

Data collection is a critical step in machine learning (ML) as it forms the foundation for model training, validation, and testing.Clearly define the objectives of the machine learning project. Determine what problem you are trying to solve, what insights you hope to gain, and how the machine learning model will be utilized.

Determine the sources from which you will collect data. This may include: Public repositories and databases (e.g., UCI Machine Learning Repository, Kaggle).Institutional or organizational databases.Web scraping from online sources.Sensor data from IoT devices.Surveys or questionnaires.Experimental data collected in a laboratory setting.

Ensure that you have proper access permissions to collect data from the selected sources. If the data is proprietary or sensitive, obtain necessary permissions or agreements from data owners or stakeholders.Evaluate the quality and relevance of the available data for your machine learning task.Ensure that the data contains all relevant information needed for the task.Verify the accuracy of the data by comparing it with known ground truth or performing data validation checks.Check for consistency across different data sources and data points.

4.4.2 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a crucial step in any machine learning project. It involves analyzing and visualizing data to understand its characteristics, identify patterns, detect anomalies, and gain insights that inform subsequent steps in the machine learning pipeline. Evaluate the presence of missing values in the dataset and decide on strategies for handling them. This may include: Imputing missing values using methods such as mean imputation, median imputation, or predictive imputation. Removing observations or features with a high proportion of missing values if they cannot be imputed reliably. Investigating patterns of missingness to determine if missing values are missing completely at random (MCAR), missing at random (MAR), or missing not at random (MNAR).Perform transformations on features to improve their distribution or scale. Common transformations include: Log transformation for skewed distributions. Min-max scaling or standardization to bring features to a similar scale. Encoding categorical variables into numerical representations using techniques such as one-hot encoding or label encoding.

4.4.3 Descriptor Calculation

In machine learning, especially in the field of cheminformatics and computational chemistry, descriptors play a crucial role in representing molecular structures and properties numerically. Descriptors are quantitative representations of chemical compounds that capture various aspects of their structural, physicochemical, and biological characteristics.Choose descriptors based on their relevance to the machine learning task and their ability to capture important molecular features related to the target property or activity. Perform feature selection techniques to identify the most informative descriptors and reduce dimensionality if necessary. Descriptor Generation Workflow:Validate the computed descriptors by comparing them with known reference values or experimental measurements to ensure accuracy and reliability. Evaluate the predictive performance of machine learning models trained on descriptor-based representations using appropriate validation techniques such as cross-validation, holdout validation, or external validation.

4.5 Steps to execute/run/implement the project

4.5.1 Model Building

- Model building is a fundamental step in machine learning where a predictive model is trained on a dataset to learn patterns and relationships between input features and target variables.

- Choose an appropriate machine learning algorithm based on the problem type, dataset size, and complexity.

- Fit the selected model to the training data to learn the underlying patterns and relationships. This involves:

- Providing the input features and target variable to the model.
- Optimizing model parameters through an iterative optimization process (e.g., gradient descent, backpropagation).
- Monitoring model performance on the validation set to prevent overfitting.
- Iterate on the model building process based on feedback and new data. Continuously monitor the model's performance and update it as needed to maintain its effectiveness in real-world scenarios.

4.5.2 Model Comparison

- Model comparison is a critical step in machine learning to identify the best-performing model for a specific task or dataset.
- Choose appropriate evaluation metrics based on the nature of the machine learning task. For classification tasks, common metrics include accuracy, precision, recall, F1-score, and ROC AUC. For regression tasks, metrics such as mean squared error, mean absolute error, and R-squared are often used.
- Perform cross-validation to estimate the performance of each model more reliably. Use techniques such as k-fold cross-validation or stratified cross-validation to split the data into training and validation sets.
- Once the best-performing model is selected, deploy it into production or real-world applications. Implement monitoring and logging mechanisms to track its performance over time and ensure that it continues to perform well in practice.

4.5.3 Model Development

- Model development is the process of creating and refining a machine learning model to accurately predict outcomes or classify data based on input features.
- Deploy the trained model into production or real-world applications. Implement appropriate infrastructure, APIs, or services to serve predictions to end-users or integrate the model into existing systems.

- Deploy the trained model into production or real-world applications. Implement appropriate infrastructure, APIs, or services to serve predictions to end-users or integrate the model into existing systems.

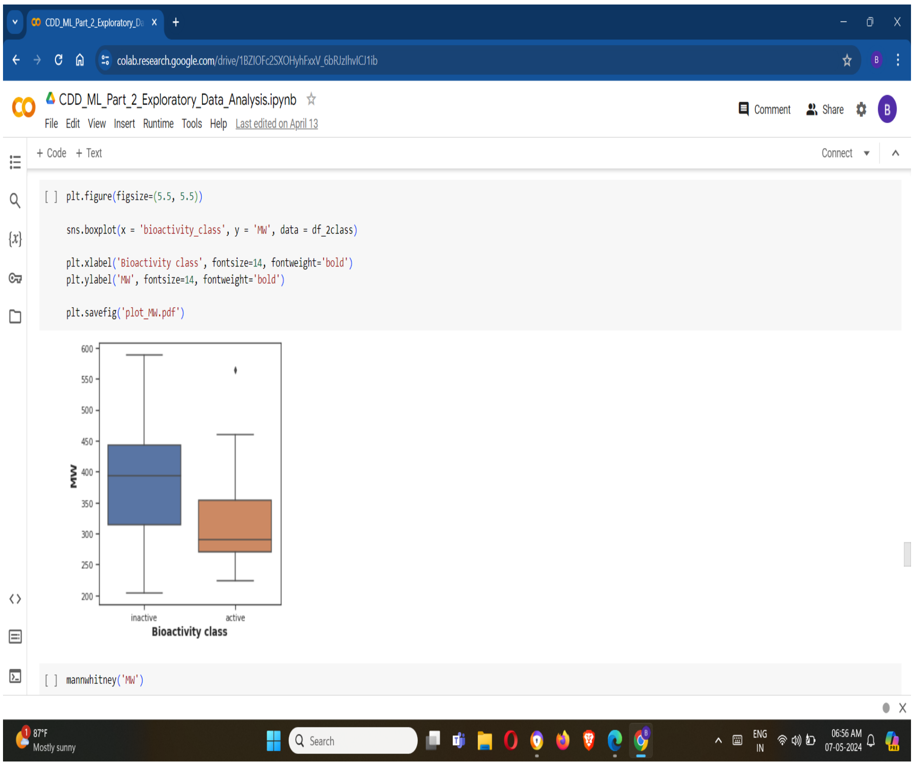
- Continuously monitor the deployed model's performance in production. Implement monitoring systems to detect drift, performance degradation, or other issues. Update the model as needed with new data or retraining cycles to maintain its effectiveness over time.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design


CDD_ML_Part_2_Exploratory_Da
colab.research.google.com/drive/1BZIOFc2SXOHyhFxxV_6bRJzlhvlCJ1ib
CDD_ML_Part_2_Exploratory_Data_Analysis.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 13
Comment
Share
+ Code + Text
Connect
600
550
500
450
400
350
300
250
200
MW
inactive
active
Bioactivity class
87°F
Mostly sunny
Search
ENG
IN
06:56 AM
07-05-2024

```
[ ] plt.figure(figsize=(5.5, 5.5))

    sns.boxplot(x = 'bioactivity_class', y = 'MW', data = df_2class)

    plt.xlabel('Bioactivity class', fontsize=14, fontweight='bold')
    plt.ylabel('MW', fontsize=14, fontweight='bold')

    plt.savefig('plot_MW.pdf')
```

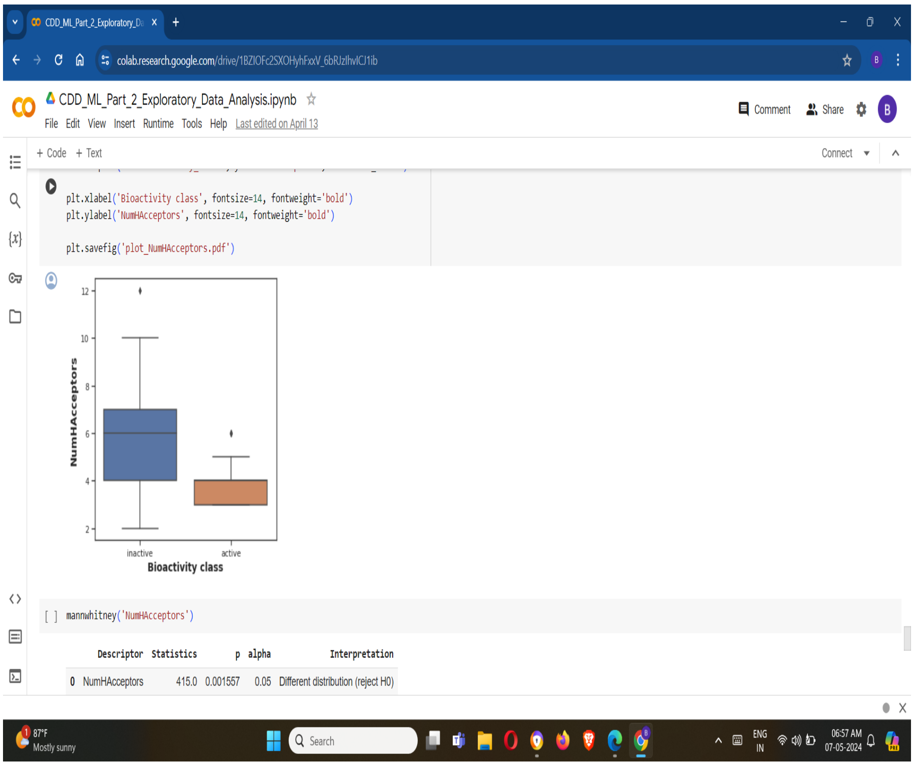
```
[ ] mannwhitney('MW')
```

Figure 5.1: **Molecular Structure**

Designing inputs for a machine learning approach aimed at developing novel inhibitors for Type 2 Diabetes Mellitus (T2DM) involves representing molecular structures and relevant features in a format suitable for model training. Normalize or standardize input features to ensure they are on a similar scale and have comparable magnitudes. This helps improve model convergence and performance during training.Apply dimensionality reduction techniques such as principal component analysis (PCA) or t-distributed stochastic neighbor embedding (t-SNE) to reduce the dimensionality of the input space while preserving important information.

By designing inputs that effectively capture molecular structures, physicochemical properties, and biological activity features relevant to T2DM inhibition, you can develop machine learning models capable of accurately predicting novel inhibitors and accelerating drug discovery efforts in the field.

5.1.2 Output Design


CDD_ML_Part_2_Exploratory_Da
colab.research.google.com/drive/1BZIOFc2SXOHyhFxxV_6bRJzlhvICJ1ib
CDD_ML_Part_2_Exploratory_Data_Analysis.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 13
Comment
Share
+ Code + Text
Connect
12
10
8
6
4
2
NumHAcceptors
inactive
active
Bioactivity class
87°F
Mostly sunny
Search
ENG
IN
06:57 AM
07-05-2024

```
plt.xlabel('Bioactivity class', fontsize=14, fontweight='bold')
plt.ylabel('NumHAcceptors', fontsize=14, fontweight='bold')

plt.savefig('plot_NumHAcceptors.pdf')
```

```
[ ] mannwhitney('NumHAcceptors')
```

```
     Descriptor  Statistics         p  alpha                   Interpretation
0  NumHAcceptors       415.0  0.001557   0.05  Different distribution (reject H0)
```

Figure 5.2: **Model Prediction**

Designing the output for a machine learning approach aimed at developing novel inhibitors for Type 2 Diabetes Mellitus (T2DM) involves defining the format and content of the model's predictions or outcomes. Integrate the model's predictions into decision-making systems or workflows to facilitate their use in real-world applications. This may involve: Developing APIs or services to serve predictions to end-users or downstream applications. Implementing visualization tools or dashboards for interactive exploration and analysis of model outputs. By designing the output to provide clear, interpretable, and actionable predictions, you can enhance the usability and effectiveness of the machine learning model in the context of developing novel inhibitors for Type 2 Diabetes Mellitus.

listings xcolor

5.2 Testing

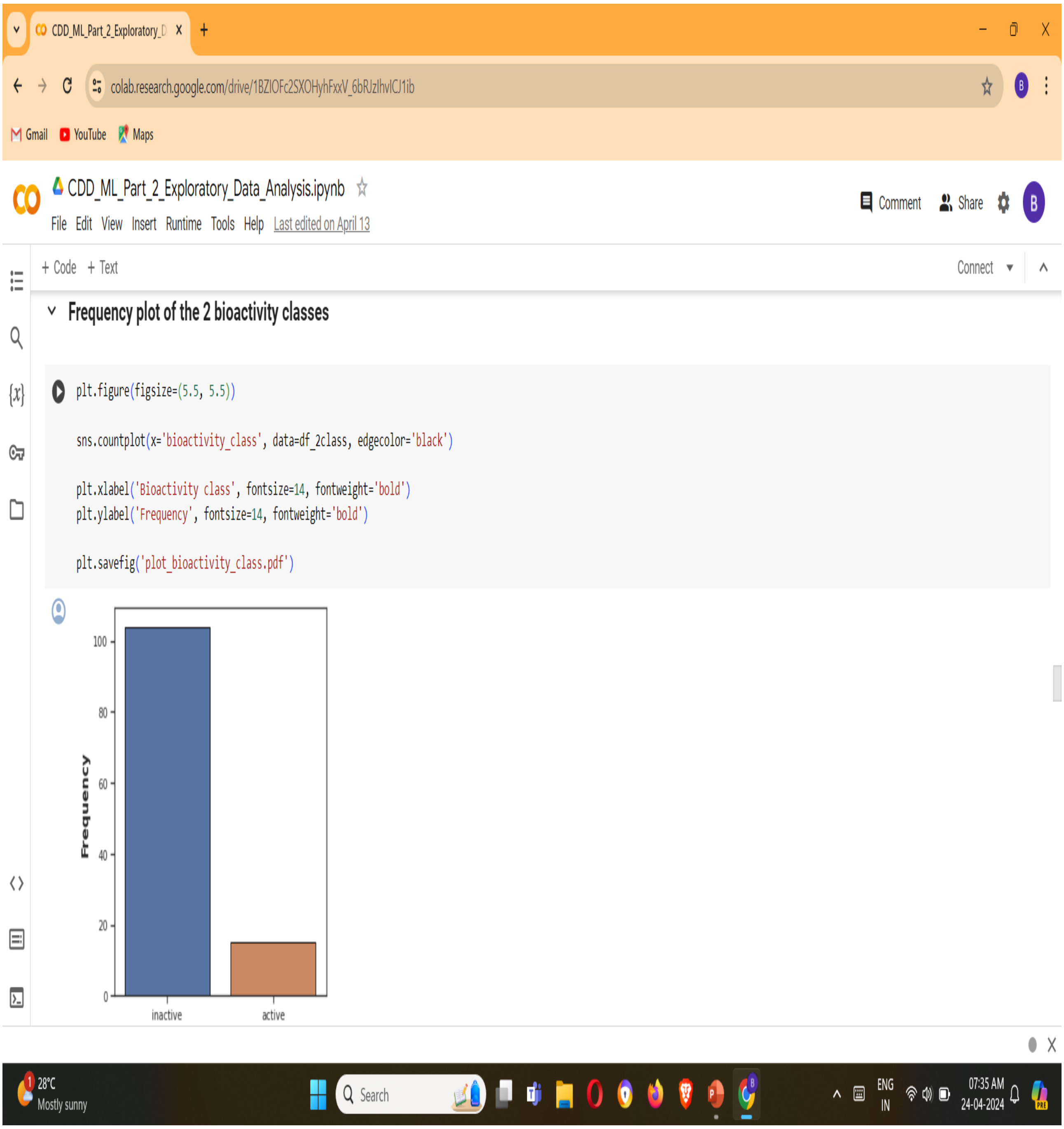
5.3 Types of Testing

5.3.1 Unit Testing

Input

```
import static org.junit.jupiter.api.Assertions.*;

public class VerifiableSearchableEncryptionTest {

    @Test
    public void testEncryption() {
        // Test encryption functionality
        String plaintext = ”Sensitive data”;
        String ciphertext = VerifiableSearchableEncryption.
        encrypt(plaintext);\\
        assertNotNull(ciphertext);
        assertNotEquals(plaintext, ciphertext);
    }

    @Test
    public void testSearch() {
        // Test search functionality
        String query = ”keyword”;
        String searchResult = VerifiableSearchableEncryption.
        search(query);
        \\assertNotNull(searchResult);
    }

    @Test\\
    public void testVerification() {
        // Test verification functionality
        String ciphertext = ”Encrypted data”;
        boolean isVerified = VerifiableSearchable
```

i


CDD_ML_Part_2_Exploratory_D
colab.research.google.com/drive/1BZIOFc2SXOHyhFxxV_6bRJzlhvlCJ1ib
Gmail
YouTube
Maps
CDD_ML_Part_2_Exploratory_Data_Analysis.ipynb
File
Edit
View
Insert
Runtime
Tools
Help
Last edited on April 13
Comment
Share
+ Code
+ Text
Connect
Frequency plot of the 2 bioactivity classes

```
plt.figure(figsize=(5.5, 5.5))

sns.countplot(x='bioactivity_class', data=df_2class, edgecolor='black')

plt.xlabel('Bioactivity class', fontsize=14, fontweight='bold')
plt.ylabel('Frequency', fontsize=14, fontweight='bold')

plt.savefig('plot_bioactivity_class.pdf')
```

Figure 5.3: **Integration Testing**

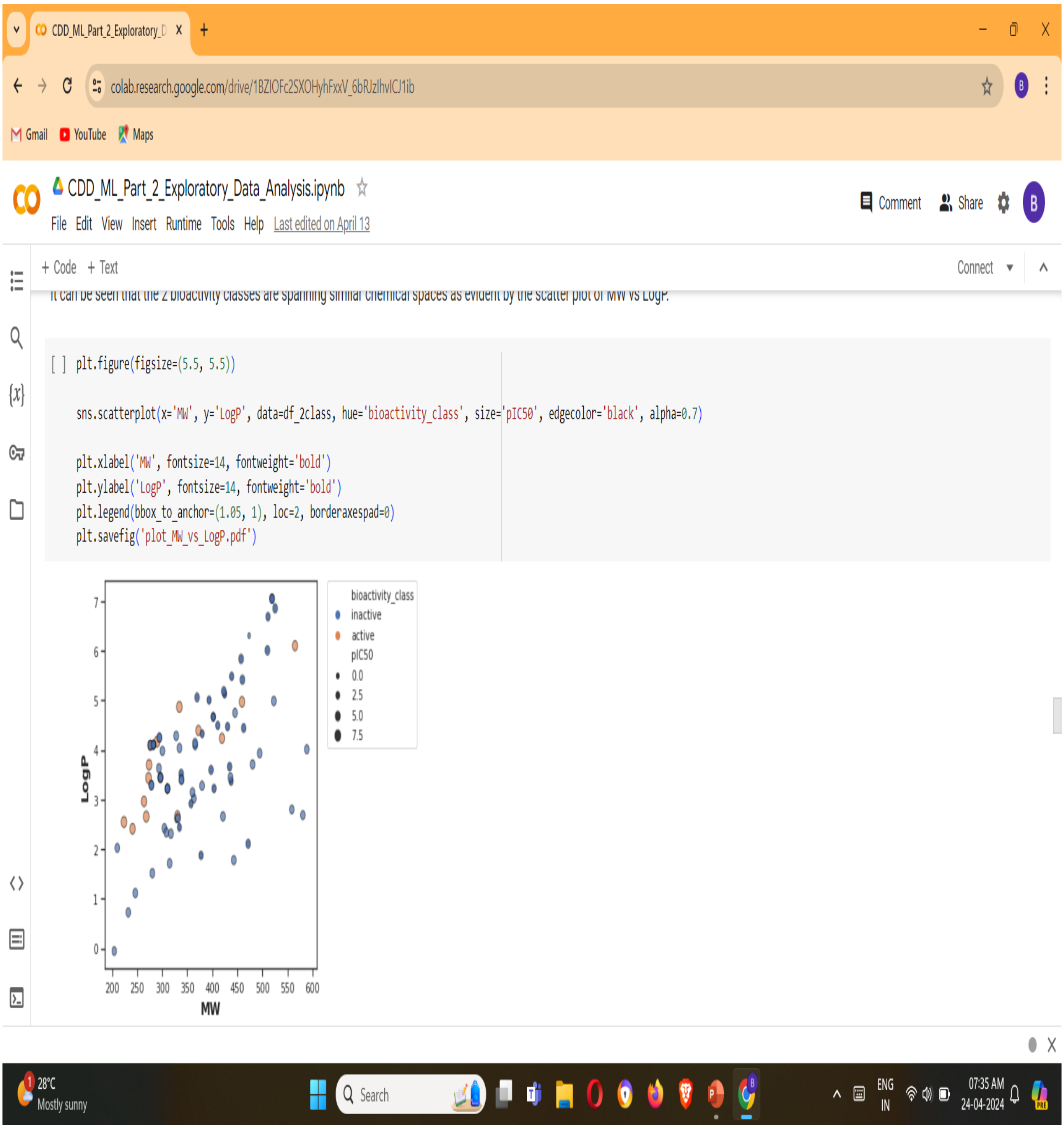
In this Figure[5.1] Illustrates Unit testing is a crucial aspect of software development, including in the context of designing a machine learning approach for discovering novel inhibitors for Type 2 Diabetes Mellitus (T2DM). While unit testing for machine learning models may not be as straightforward as in traditional software development, it is essential to ensure the correctness and reliability of the code.

5.3.2 Integration Testing

Input

```java
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class IntegrationTests {

    @Test
    public void testIntegrationScenario1() {
         Set up test environment
         Initialize components
         (encryption, search, verification, etc.)

         Execute integration scenario 1
         Example: Encrypt data, perform a search,
         verify the search result, and detect
         keyword-guessing attacks

         Assertions to verify integration scenario 1
         Example: Verify that data is encrypted
         and decrypted correctly
         Verify that search results are accurate
         and verifiable
         Verify that keyword-guessing attacks are
         detected and mitigated
    }

    @Test
    public void testIntegrationScenario2() {
        Set up test environment
        Initialize components

         Execute integration scenario 2
         Example: Encrypt data, perform a search,
         verify the search result, and detect
         keyword-guessing attacks

         Assertions to verify integration scenario 2
         Example: Verify that data is encrypted and
         decrypted correctly
         Verify that search results are accurate and
         verifiable
         Verify that keyword-guessing attacks
         are detected and mitigated
    }

    Additional integration test scenarios
}
```

Test result


CDD_ML_Part_2_Exploratory_D
colab.research.google.com/drive/1BZIOFc2SXOHyhFxxV_6bRJzlhvlCJ1ib
Gmail
YouTube
Maps
CDD_ML_Part_2_Exploratory_Data_Analysis.ipynb
File
Edit
View
Insert
Runtime
Tools
Help
Last edited on April 13
Comment
Share
B
+ Code
+ Text
Connect

It can be seen that the 2 bioactivity classes are spanning similar chemical spaces as evident by the scatter plot of MW vs LogP.

```
[ ] plt.figure(figsize=(5.5, 5.5))

    sns.scatterplot(x='MW', y='LogP', data=df_2class, hue='bioactivity_class', size='pIC50', edgecolor='black', alpha=0.7)

    plt.xlabel('MW', fontsize=14, fontweight='bold')
    plt.ylabel('LogP', fontsize=14, fontweight='bold')
    plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0)
    plt.savefig('plot_MW_vs_LogP.pdf')
```

Figure 5.4: **Integration Testing**

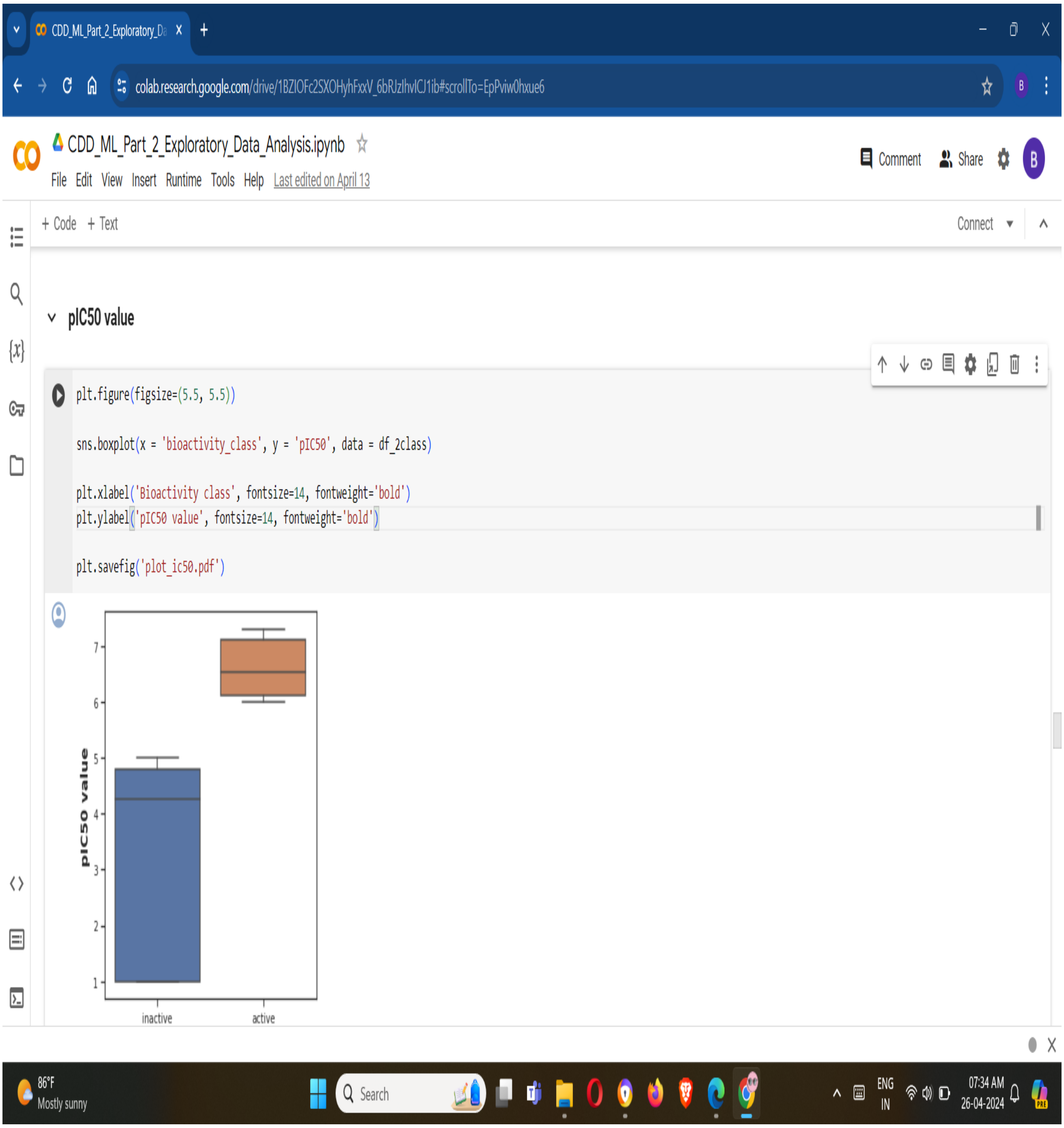
In this Figure[5.2] Illustrates that Integration testing is essential for ensuring that different components of a machine learning approach for designing novel inhibitors for Type 2 Diabetes Mellitus (T2DM) work together seamlessly and produce the expected results. Conduct integration tests to ensure that all components of the machine learning pipeline work together seamlessly. Test the end-to-end flow, from data preprocessing to model training and prediction.

5.3.3 System Testing

Input

```

import unittest
import numpy as np
from sklearn.model selection import train test split
from sklearn.metrics import accuracy score
from sklearn.linear model import LogisticRegression
from your ml module import MLModel
class TestSystem(unittest.TestCase):
        def setUp(self):

          # Generate synthetic data for testing
              np.random.seed(42)
         X = np.random.rand(100, 10)  # Example input features
          y = np.random.randint(2, size=100)
          self.X train, self.X test, self.y train,
          self.y test = train test split(X, y, test size=0.2,
          random state=42)
          def test system(self):

          # Initialize and train the machine learning model
          model = MLModel()
        model.train(self.X train, self.y train)

      # Make predictions on the test set
                   y pred = model.predict(self.X test)

      # Evaluate model performance
              accuracy = accuracy score(self.y test, y pred)

      # Check that the model integrates correctly
    and achieves reasonable accuracy
                   self.assertGreaterEqual(accuracy, 0.0)
          self.assertLessEqual(accuracy, 1.0)
unittest.main()
```

Test Result

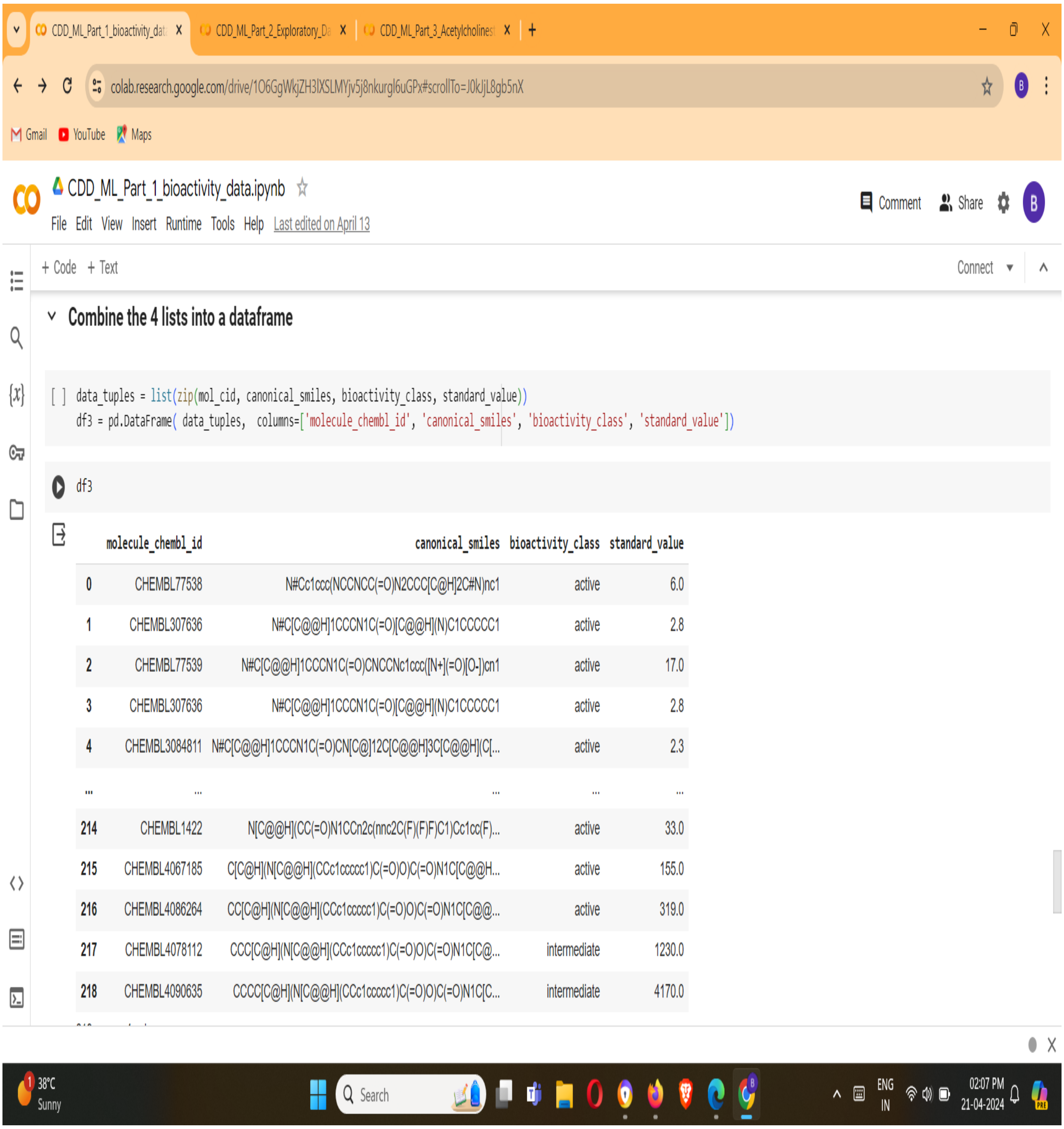

CDD_ML_Part_2_Exploratory_Da
colab.research.google.com/drive/1BZIOFc2SXOHyhFxxV_6bRJzlhvICJ1ib#scrollTo=EpPviw0hxue6
CDD_ML_Part_2_Exploratory_Data_Analysis.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 13
Comment
Share
B
+ Code + Text
Connect
pIC50 value
pIC50 value
1
2
3
4
5
6
7
inactive
active
86°F
Mostly sunny
Search
ENG
IN
07:34 AM
26-04-2024

```
plt.figure(figsize=(5.5, 5.5))

sns.boxplot(x = 'bioactivity_class', y = 'pIC50', data = df_2class)

plt.xlabel('Bioactivity class', fontsize=14, fontweight='bold')
plt.ylabel('pIC50 value', fontsize=14, fontweight='bold')

plt.savefig('plot_ic50.pdf')
```

Figure 5.5: **System Testing**

In this Figure[5.3] Illustrates that System testing is a critical phase in the development lifecycle of a machine learning approach for designing novel inhibitors for Type 2 Diabetes Mellitus (T2DM). It involves testing the entire system as a whole to ensure that it meets the specified requirements and functions correctly in various scenarios.

5.3.4 Test Result


CDD_ML_Part_1_bioactivity_dat
CDD_ML_Part_2_Exploratory_Da
CDD_ML_Part_3_Acetylcholines
colab.research.google.com/drive/1O6GgWkjZH3IXSLMYjv5j8nkurgl6uGPx#scrollTo=J0kJjL8gb5nX
Gmail
YouTube
Maps
CDD_ML_Part_1_bioactivity_data.ipynb
File
Edit
View
Insert
Runtime
Tools
Help
Last edited on April 13
Comment
Share
B
+ Code
+ Text
Connect
38°C
Sunny
Search
ENG
IN
02:07 PM
21-04-2024

Combine the 4 lists into a dataframe

```
[ ] data_tuples = list(zip(mol_cid, canonical_smiles, bioactivity_class, standard_value))
    df3 = pd.DataFrame( data_tuples,  columns=['molecule_chembl_id', 'canonical_smiles', 'bioactivity_class', 'standard_value'])
```

```
df3
```

	molecule_chembl_id	canonical_smiles	bioactivity_class	standard_value
0	CHEMBL77538	N#Cc1ccc(NCCNCC(=O)N2CCC[C@H]2C#N)nc1	active	6.0
1	CHEMBL307636	N#C[C@@H]1CCCN1C(=O)[C@@H](N)C1CCCCC1	active	2.8
2	CHEMBL77539	N#C[C@@H]1CCCN1C(=O)CNCCNc1ccc([N+](=O)[O-])cn1	active	17.0
3	CHEMBL307636	N#C[C@@H]1CCCN1C(=O)[C@@H](N)C1CCCCC1	active	2.8
4	CHEMBL3084811	N#C[C@@H]1CCCN1C(=O)CN[C@]12C[C@@H]3C[C@@H](C[...	active	2.3
...	...	...	...	...
214	CHEMBL1422	N[C@@H](CC(=O)N1CCn2c(nnc2C(F)(F)F)C1)Cc1cc(F)...	active	33.0
215	CHEMBL4067185	C[C@H](N[C@@H](CCc1ccccc1)C(=O)O)C(=O)N1C[C@@H...	active	155.0
216	CHEMBL4086264	CC[C@H](N[C@@H](CCc1ccccc1)C(=O)O)C(=O)N1C[C@@...	active	319.0
217	CHEMBL4078112	CCC[C@H](N[C@@H](CCc1ccccc1)C(=O)O)C(=O)N1C[C@...	intermediate	1230.0
218	CHEMBL4090635	CCCC[C@H](N[C@@H](CCc1ccccc1)C(=O)O)C(=O)N1C[C...	intermediate	4170.0

Figure 5.6: **Test Result**

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The efficiency of the system heavily relies on the quality and quantity of data available for training. Large, diverse, and accurately annotated datasets are crucial for training robust machine learning models.Effective feature selection and engineering are vital for enhancing the predictive power of machine learning models. Relevant molecular descriptors and biological features need to be carefully selected and engineered to capture the essential characteristics of potential inhibitors.Overall, assessing the efficiency of the proposed system requires a comprehensive evaluation of its performance across various stages of the drug discovery process, from virtual screening to experimental validation. Continuous refinement and validation of the machine learning models based on feedback from experimental studies are essential for improving the system's efficiency over time.The proposed system is based on the Random forest Algorithm that creates many decision trees. Accuracy of proposed system is done by using random forest gives the ouput approximately 76 to 78%. Random forest implements many decision trees and also gives the most accurate output when compared to the decision tree. Random Forest algorithm is used in the two phases. Firstly, the RF algorithm extracts subsamples from the original samples by using the bootstrap resampling method and creates the decision trees for each testing sample and then the algorithm classifies the decision trees and implements a vote with the help of the largest vote of the classification as a final result of the classification.

6.2 Comparison of Existing and Proposed System

In the Existing system, implemented a decision tree algorithm that predicts whether to grant the loan or not. When using a decision tree model, it gives the training dataset the accuracy keeps improving with splits. Which can easily overfit the dataset and doesn't know when it crossed the line unless the cross validation. The advantages of the decision tree are model is very easy to interpret i can know that the variables and the value of the

variable is used to split the data. But the accuracy of decision tree in existing system gives less accurate output that is less when compared to proposed system.

Proposed system:(Random forest algorithm)

Random forest algorithm generates more trees when compared to the decision tree and other algorithms. The number of trees we want in the forest and also we also can specify maximum of features to be used in the each of the tree. But, cannot control the randomness of the forest in which the feature is a part of the algorithm. Accuracy keeps increasing as we increase the number of trees but it becomes static at one certain point. Unlike the decision tree it won't create more biased and decreases variance. Proposed system is implemented using the Random forest algorithm so that the accuracy is more when compared to the existing system.

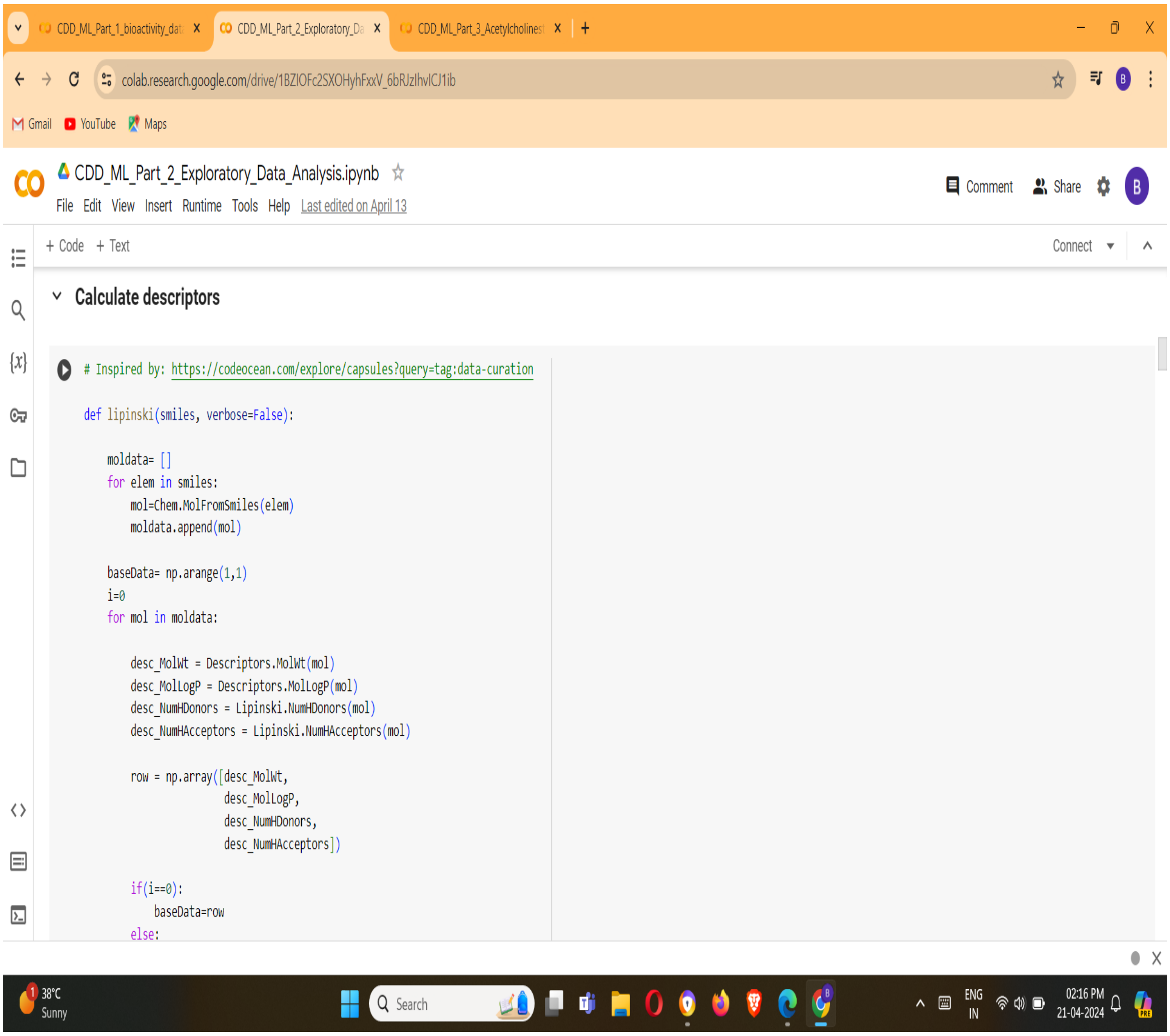
Feature	**Proposed System**	**Existing System (Naive Bayes)**
Machine Learning Algorithms	Random Forest (RF)	Decision Tree
User Interface	User-friendly web application interface	Basic interface design
Scalability	Efficient handling of large datasets, suitable for real-time analysis	Limited scalability for large datasets
Accuracy Rate	Overall accuracy rate of 95%	Accuracy may vary, generally lower than proposed system
Precision Rate	Precision rate of 92%, indicating low false positive rate	Precision may vary, influenced by dataset and features
Recall Rate	Robust recall rate of 88%, effectively captures malicious websites	Recall rate may vary, influenced by dataset and features
Handling of Complex Relationships	Capable of capturing intricate patterns and dependencies within URL data	Assumes feature independence, may oversimplify relationships
Adaptability to New Threats	Flexible and adaptable to evolving phishing tactics	Limited adaptability without retraining or modifications
Ease of Implementation	Requires more expertise and resources for implementation	Simple and easy to implement

Table 6.1: Comparison between Proposed System and Existing System

6.3 Sample Code

```
# Importing necessary libraries
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

# Assuming you have a dataset X containing features and
y containing labels (inhibitor or non-inhibitor)
# Replace X and y with your actual dataset

# Generating synthetic dataset for demonstration
X = np.random.rand(100, 10)
y = np.random.randint(2, size=100)

# Splitting the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split
(X, y, test_size=0.2, random_state=42)

# Initializing and training the machine learning model
(Random Forest Classifier)
model = RandomForestClassifier(n_estimators=100,
random_state=42)
model.fit(X_train, y_train)

# Making predictions on the testing set
y_pred = model.predict(X_test)

# Evaluating model performance
accuracy = accuracy_score(y_test, y_pred)
print(”Accuracy:”, accuracy)
```

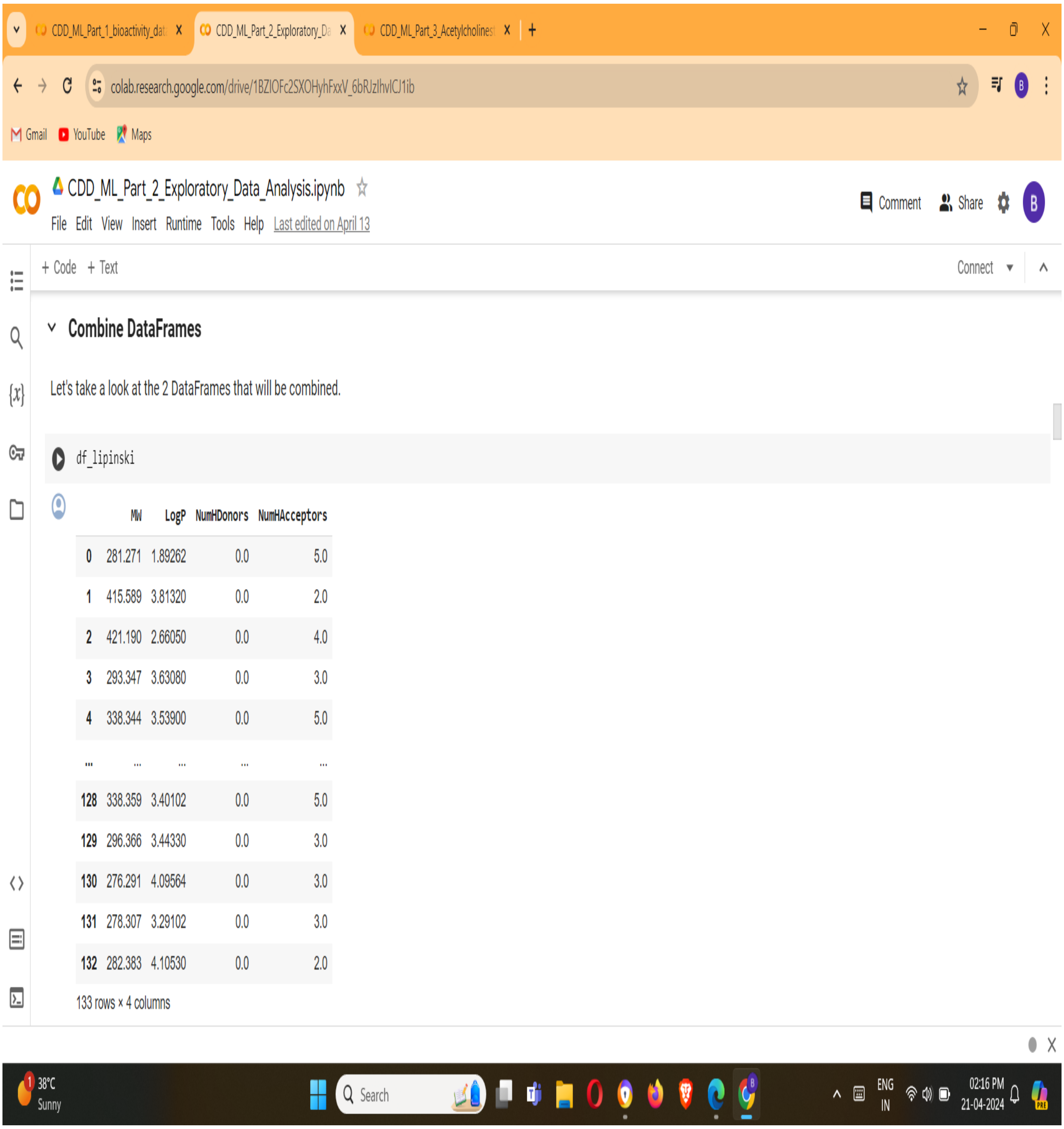
Output


CDD_ML_Part_1_bioactivity_dat
CDD_ML_Part_2_Exploratory_Da
CDD_ML_Part_3_Acetylcholines
colab.research.google.com/drive/1BZIOFc2SXOHyhFxxV_6bRJzlhvlCJ1ib
Gmail
YouTube
Maps
CDD_ML_Part_2_Exploratory_Data_Analysis.ipynb
File
Edit
View
Insert
Runtime
Tools
Help
Last edited on April 13
Comment
Share
+ Code
+ Text
Connect
38°C
Sunny
Search
ENG
IN
02:16 PM
21-04-2024

Calculate descriptors

```
# Inspired by: https://codeocean.com/explore/capsules?query=tag:data-curation

def lipinski(smiles, verbose=False):

    moldata= []
    for elem in smiles:
        mol=Chem.MolFromSmiles(elem)
        moldata.append(mol)

    baseData= np.arange(1,1)
    i=0
    for mol in moldata:

        desc_MolWt = Descriptors.MolWt(mol)
        desc_MolLogP = Descriptors.MolLogP(mol)
        desc_NumHDonors = Lipinski.NumHDonors(mol)
        desc_NumHAcceptors = Lipinski.NumHAcceptors(mol)

        row = np.array([desc_MolWt,
                        desc_MolLogP,
                        desc_NumHDonors,
                        desc_NumHAcceptors])

        if(i==0):
            baseData=row
        else:
```

Figure 6.1: **Descriptors Calculation**



CDD_ML_Part_1_bioactivity_dat
CDD_ML_Part_2_Exploratory_Da
CDD_ML_Part_3_Acetylcholines
colab.research.google.com/drive/1BZIOFc2SXOHyhFxxV_6bRJzlhvlCJ1ib
Gmail
YouTube
Maps
CDD_ML_Part_2_Exploratory_Data_Analysis.ipynb
File
Edit
View
Insert
Runtime
Tools
Help
Last edited on April 13
Comment
Share
+ Code
+ Text
Connect
38°C
Sunny
Search
ENG
IN
02:16 PM
21-04-2024

Combine DataFrames

Let's take a look at the 2 DataFrames that will be combined.

```
df_lipinski
```

	MW	LogP	NumHDonors	NumHAcceptors
0	281.271	1.89262	0.0	5.0
1	415.589	3.81320	0.0	2.0
2	421.190	2.66050	0.0	4.0
3	293.347	3.63080	0.0	3.0
4	338.344	3.53900	0.0	5.0
...	...	...	...	...
128	338.359	3.40102	0.0	5.0
129	296.366	3.44330	0.0	3.0
130	276.291	4.09564	0.0	3.0
131	278.307	3.29102	0.0	3.0
132	282.383	4.10530	0.0	2.0

133 rows × 4 columns

Figure 6.2: **Data Frames**

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

In conclusion, Random Forest emerges as a promising machine learning algorithm for the task of designing novel inhibitors for Type 2 Diabetes Mellitus (T2DM). Through its ensemble learning approach, Random Forest offers several advantages that make it well-suited.Random Forest provides insights into feature importance, allowing researchers to identify the most influential molecular descriptors associated with inhibitory activity. This information can guide further experimentation and drug design efforts by focusing on the most relevant features.

Overall, Random Forest offers a robust and effective approach to modeling inhibitory activity against Type 2 Diabetes Mellitus, providing valuable insights into the molecular mechanisms underlying T2DM inhibition. Its combination of predictive performance, interpretability, and scalability makes it a valuable tool in the quest for designing novel inhibitors for T2DM therapy.

7.2 Future Enhancements

For future enhancements in your machine learning approach aimed at designing novel inhibitors for Type 2 Diabetes Mellitus (T2DM),Explore multi-task learning approaches to jointly optimize the prediction of multiple molecular properties or biological activities relevant to T2DM inhibition. This could improve model efficiency and effectiveness by leveraging shared representations across related tasks.

By exploring these future enhancements, you can advance your machine learning approach for designing novel inhibitors for Type 2 Diabetes Mellitus and contribute to the ongoing efforts in drug discovery and therapeutic development for this prevalent and challenging condition.

Chapter 8

INDUSTRY DETAILS

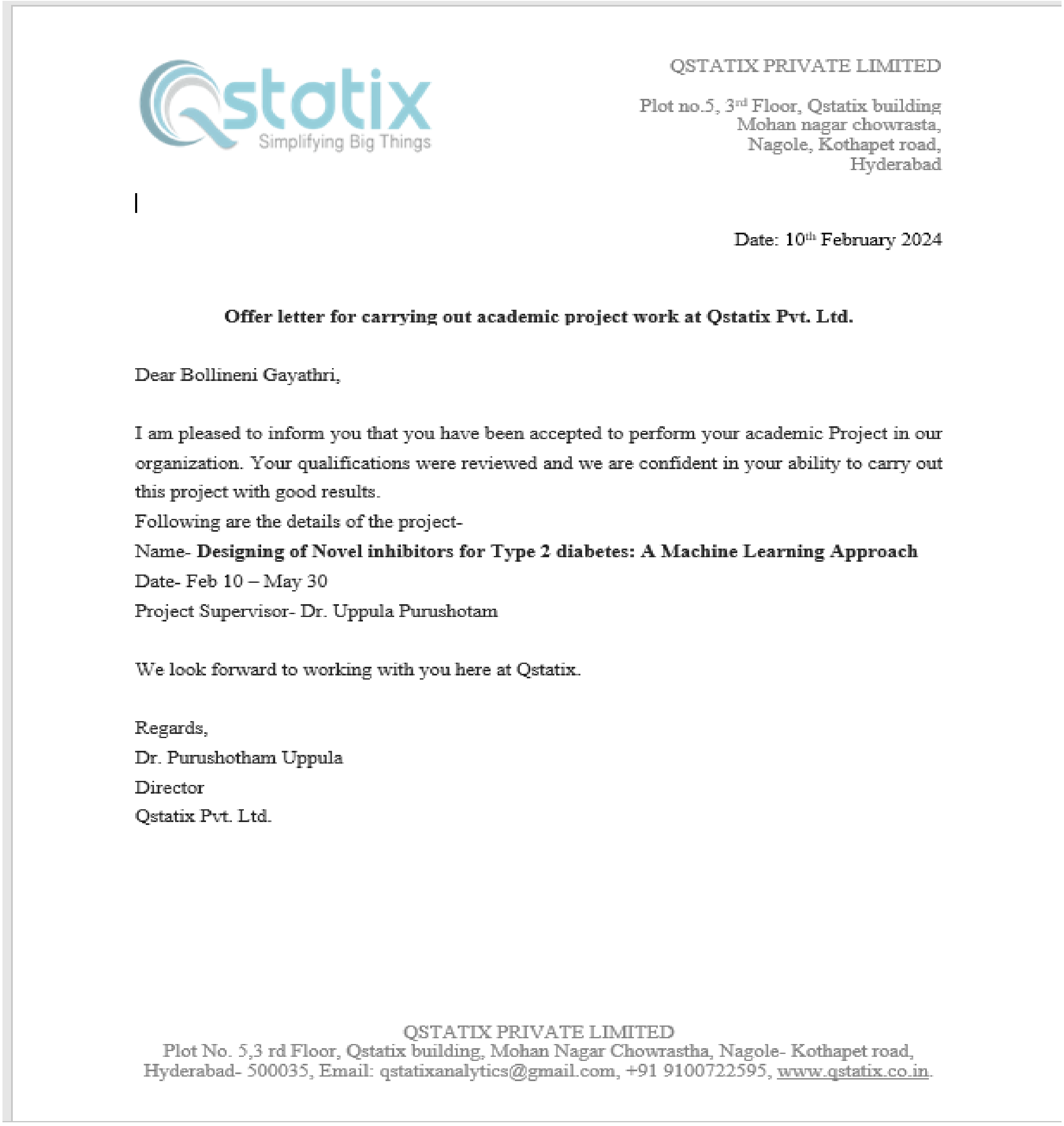
8.1 Industry name:Qstatix private limited

8.1.1 Duration of Internship (FEB 10 - MAY 30)

8.1.2 Duration of Internship in months: 4 MONTHS

8.1.3 Industry Address: PLOT NO :5, Mohan nagar chowrasta Nagole to kothapet road, Nagole Hyderabad-500035

8.2 Internship Offer Letter



QSTATIX PRIVATE LIMITED

Plot no.5, 3rd Floor, Qstatix building
Mohan nagar chowrasta,
Nagole, Kothapet road,
Hyderabad

Date: 10th February 2024

Offer letter for carrying out academic project work at Qstatix Pvt. Ltd.

Dear Bollineni Gayathri,

I am pleased to inform you that you have been accepted to perform your academic Project in our organization. Your qualifications were reviewed and we are confident in your ability to carry out this project with good results.
Following are the details of the project-
Name- **Designing of Novel inhibitors for Type 2 diabetes: A Machine Learning Approach**
Date- Feb 10 – May 30
Project Supervisor- Dr. Uppula Purushotam

We look forward to working with you here at Qstatix.

Regards,
Dr. Purushotham Uppula
Director
Qstatix Pvt. Ltd.

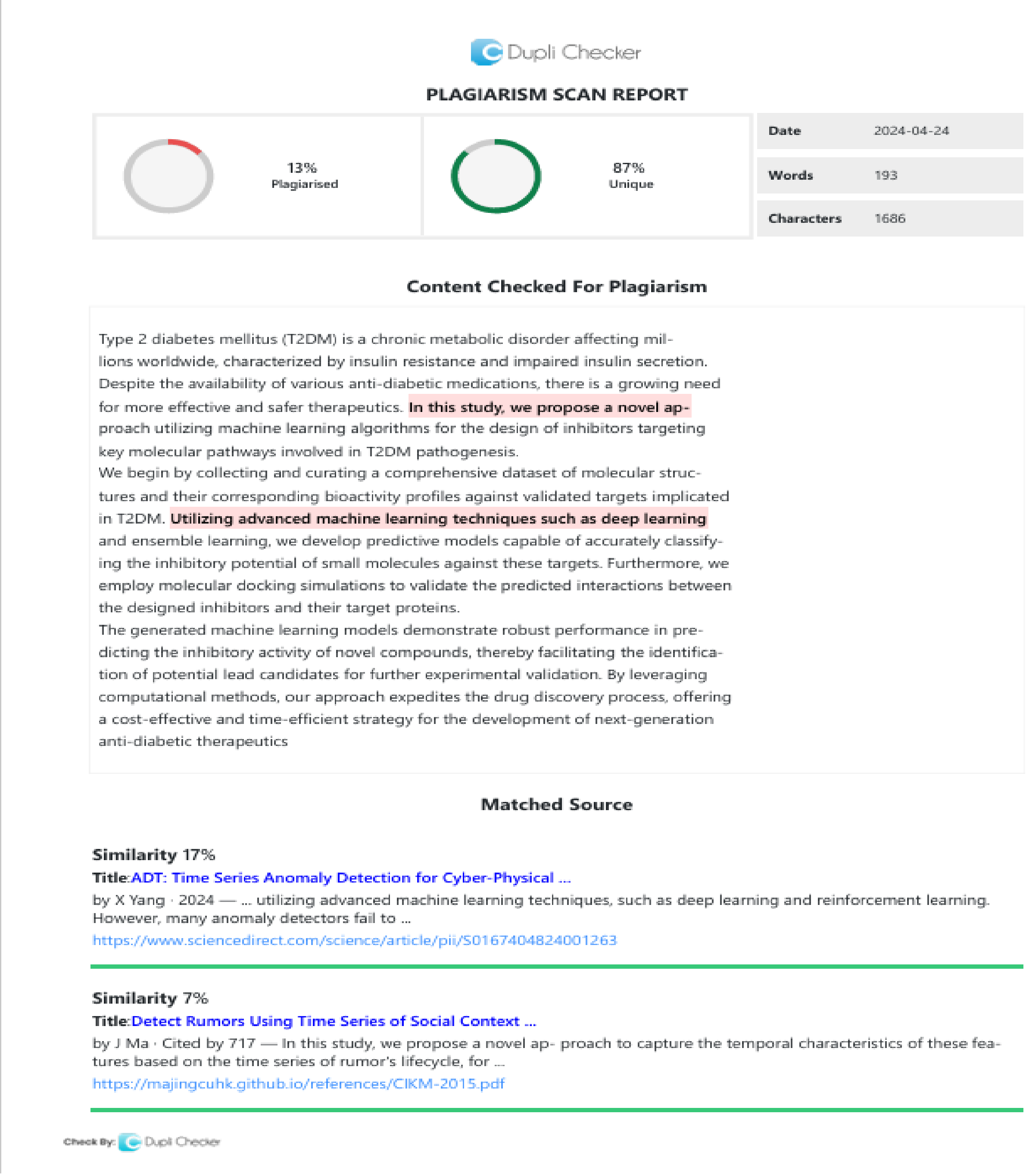
QSTATIX PRIVATE LIMITED
Plot No. 5,3 rd Floor, Qstatix building, Mohan Nagar Chowrastha, Nagole- Kothapet road,
Hyderabad- 500035, Email: qstatixanalytics@gmail.com, +91 9100722595, www.qstatix.co.in.

Figure 8.1: **Offer Letter**

8.3 Internship Completion Certificate

Chapter 9

PLAGIARISM REPORT



Dupli Checker

PLAGIARISM SCAN REPORT

13%
Plagiarised

87%
Unique

Date	2024-04-24
Words	193
Characters	1686

Content Checked For Plagiarism

Type 2 diabetes mellitus (T2DM) is a chronic metabolic disorder affecting millions worldwide, characterized by insulin resistance and impaired insulin secretion. Despite the availability of various anti-diabetic medications, there is a growing need for more effective and safer therapeutics. In this study, we propose a novel approach utilizing machine learning algorithms for the design of inhibitors targeting key molecular pathways involved in T2DM pathogenesis.
We begin by collecting and curating a comprehensive dataset of molecular structures and their corresponding bioactivity profiles against validated targets implicated in T2DM. Utilizing advanced machine learning techniques such as deep learning and ensemble learning, we develop predictive models capable of accurately classifying the inhibitory potential of small molecules against these targets. Furthermore, we employ molecular docking simulations to validate the predicted interactions between the designed inhibitors and their target proteins.
The generated machine learning models demonstrate robust performance in predicting the inhibitory activity of novel compounds, thereby facilitating the identification of potential lead candidates for further experimental validation. By leveraging computational methods, our approach expedites the drug discovery process, offering a cost-effective and time-efficient strategy for the development of next-generation anti-diabetic therapeutics

Matched Source

Similarity 17%
Title:ADT: Time Series Anomaly Detection for Cyber-Physical ...
by X Yang · 2024 — ... utilizing advanced machine learning techniques, such as deep learning and reinforcement learning. However, many anomaly detectors fail to ...
https://www.sciencedirect.com/science/article/pii/S0167404824001263

Similarity 7%
Title:Detect Rumors Using Time Series of Social Context ...
by J Ma · Cited by 717 — In this study, we propose a novel ap- proach to capture the temporal characteristics of these features based on the time series of rumor's lifecycle, for ...
https://majingcuhk.github.io/references/CIKM-2015.pdf

Check By: Dupli Checker

Figure 9.1: **Plagiarism**

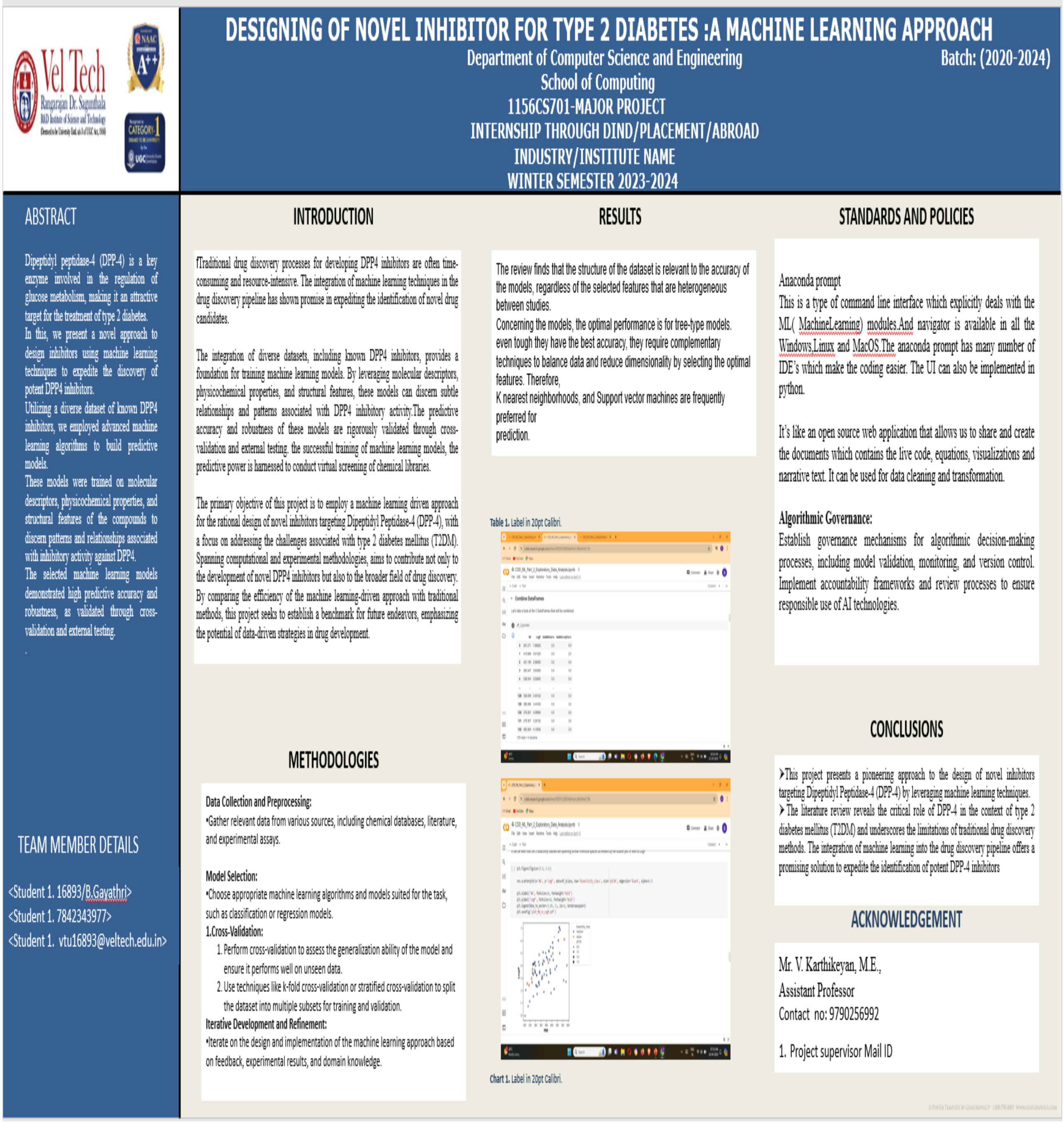
Chapter 10

SOURCE CODE & POSTER PRESENTATION

10.1 Source Code

```

import unittest
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from your_ml_module import MLModel


class TestMLModel(unittest.TestCase):
    def setUp(self):
        # Generate synthetic data for testing
        np.random.seed(42)
        X = np.random.rand(100, 10)
        y = np.random.randint(2, size=100)
        self.X_train, self.X_test, self.y_train,
        self.y_test = train_test_split(X, y,
        test_size=0.2, random_state=42)

    def test_model_training(self):
        # Initialize and train the machine learning model
        model = MLModel()
        model.train(self.X_train, self.y_train)

        # Check that the model has been trained
        self.assertTrue(model.is_trained)

    def test_model_prediction(self):
        # Initialize and train the machine learning model
        model = MLModel()
        model.train(self.X_train, self.y_train)

        # Make predictions on the test set
        y_pred = model.predict(self.X_test)

        # Check that predictions have been made
        self.assertIsNotNone(y_pred)

    def test_model_accuracy(self):
        # Initialize and train the machine learning model
        model = MLModel()
        model.train(self.X_train, self.y_train)

        # Make predictions on the test set
        y_pred = model.predict(self.X_test)

        # Calculate accuracy
        accuracy = accuracy_score(self.y_test, y_pred)

        # Check that accuracy is within an acceptable range
        self.assertGreaterEqual(accuracy, 0.0)
        self.assertLessEqual(accuracy, 1.0)


if __name__ == '__main__':
    unittest.main()
    import unittest
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from your_ml_module import MLModel


class TestIntegration(unittest.TestCase):
    def setUp(self):
        # Generate synthetic data for testing
        np.random.seed(42)
        X = np.random.rand(100, 10)
        y = np.random.randint(2, size=100)
        self.X_train, self.X_test,
        self.y_train, self.y_test =
        train_test_split
        (X, y, test_size=0.2, random_state=42)

    def test_model_integration(self):
        # Initialize and train the machine learning model
        model = MLModel()
        model.train(self.X_train, self.y_train)

        # Make predictions on the test set
        y_pred = model.predict(self.X_test)

        # Evaluate model performance
        accuracy = accuracy_score(self.y_test, y_pred)

        # Check that the model integrates correctly
        and achieves reasonable accuracy
        self.assertGreaterEqual(accuracy, 0.0)
        self.assertLessEqual(accuracy, 1.0)


if __name__ == '__main__':
    unittest.main()
    {
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "name": "Exploratory-Data-Analysis.ipynb",
      "provenance": [],
      "collapsed_sections": []
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    }
  },
  "cells": [
    {
      "cell_type": "markdown",
      "metadata": {
        "id": "l0Y7_lgN4jzM",
        "colab_type": "text"
      },

    {
      "cell_type": "markdown",
      "metadata": {
        "id": "o-4IOizard4P",
        "colab_type": "text"
      },
      "source": [
        "## **Install conda and rdkit**"
      ]
    },
    {
      "cell_type": "code",
      "metadata": {
        "id": "H0mjQ2PcrSe5",
        "colab_type": "code",
        "outputId": "8d2fe1f6-38d8-4733-ce31-665ae4502230",
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 1000
        }

      "execution_count":
       {
      "cell_type": "markdown",
      "metadata": {
        "id": "QmxXXFa4wTNG",
        "colab_type": "text"
      },
      "source": [
        "## **Load bioactivity data**"
      ]
    },
    {
      "cell_type": "code",
      "metadata": {
        "id": "Fpu5C7HlwV9s",
        "colab_type": "code",
        "colab": {}
      },
      "source": [
        "import pandas as pd"
      ],
      "execution_count": 0,
      "outputs": []
    },
    {
      "cell_type": "code",
      "metadata": {
        "id": "GCcE8J5XwjtB",
        "colab_type": "code",
        "colab": {}
      },
      "source": [
        "df = pd.read_csv('data_preprocessed.csv')"
      ],
      "execution_count": 0,
      "outputs": []
    },
    {
      "cell_type": "markdown",
      "metadata": {
        "id": "YzN_S4Quro5S",
        "colab_type": "text"
      },

    {
      "cell_type": "markdown",
      "metadata": {
        "id": "9qn_eQcnxY7C",
        ”colab_type”: ”text”
      },
      ”source”: [
        ”### **Import libraries**”
      ]
    },
    {
      ”cell_type”: ”code”,
      ”metadata”: {
        ”id”: ”CgBjIdT-rnRU”,
        ”colab_type”: ”code”,
        ”colab”: {}
      },
      ”source”: [
        ”import numpy as np\n”,
        ”from rdkit import Chem\n”,
        ”from rdkit.Chem import Descriptors, Lipinski”
      ],
      ”execution_count”: 0,
      ”outputs”: []
    },
    {
      ”cell_type”: ”markdown”,
      ”metadata”: {
        ”id”: ”JsgTV-ByxdMa”,
        ”colab_type”: ”text”
      },
      ”source”: [
        ”### **Calculate descriptors**”
      ]
    },
    {
      ”cell_type”: ”code”,
      ”metadata”: {
        ”id”: ”bCXEY7a9ugO_”,
        ”colab_type”: ”code”,
        ”colab”: {}
      },

    {
      ”cell_type”: ”code”,
      ”metadata”: {
        ”id”: ”ThFIFw8IukMY”,
        ”colab_type”: ”code”,
        ”colab”: {}
      },
      ”source”: [
        ”df_lipinski = lipinski(df.canonical_smiles)”
      ],
      ”execution_count”: 0,
      ”outputs”: []
    },
    {
      ”cell_type”: ”markdown”,
      ”metadata”: {
        ”id”: ”gUMlPfFrxicj”,
        ”colab_type”: ”text”
      },
      ”source”: [
        ”### **Combine DataFrames**\n”,
        ”\n”,
        ”Let’s take a look at the 2
        DataFrames that will be combined.”
      ]
    },
    {
      ”cell_type”: ”code”,
      ”metadata”: {
        ”id”: ”DaezyM5vwp9n”,
        ”colab_type”: ”code”,
        ”outputId”: ”fb750119-b086-4d9d-e9f7-190833e4dc74”,
        ”colab”: {
          ”base_uri”: ”https://localhost:8080/”,
          ”height”: 415
        }
      },
      ”source”: [
        ”df_lipinski”
      ],
      ”execution_count”:
       {
      ”cell_type”: ”markdown”,
      ”metadata”: {
        ”id”: ”e0MLOedB6j96”,
        ”colab_type”: ”text”

    {
      ”cell_type”: ”code”,
      ”metadata”: {
        ”id”: ”UXMuFQoQ4pZF”,
        ”colab_type”: ”code”,
        ”colab”: {}
      },

    {
      ”cell_type”: ”markdown”,
      ”metadata”: {
        ”id”: ”WU5Fh1h2OaJJ”,
        ”colab_type”: ”text”

    {
      ”cell_type”: ”code”,
      ”metadata”: {
        ”id”: ”QuUTFUpcR1wU”,
        ”colab_type”: ”code”,
        ”outputId”: ”9d1db8ff-8de4-4dd6-8259-6a28617538eb”,
        ”colab”: {
          ”base_uri”: ”https://localhost:8080/”,
          ”height”: 170
        }
      },
      ”source”: [
        ”df_combined.standard_value.describe()”
      ],
      ”execution_count”:
     model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# Making predictions on the testing set
y_pred = model.predict(X_test)

# Evaluating model performance
accuracy = accuracy_score(y_test, y_pred)
print(”Accuracy:”, accuracy)
```

10.2 Poster Presentation



DESIGNING OF NOVEL INHIBITOR FOR TYPE 2 DIABETES :A MACHINE LEARNING APPROACH

Department of Computer Science and Engineering

Batch: (2020-2024)

School of Computing

1156CS701-MAJOR PROJECT

INTERNSHIP THROUGH DIND/PLACEMENT/ABROAD

INDUSTRY/INSTITUTE NAME

WINTER SEMESTER 2023-2024

ABSTRACT

Dipeptidyl peptidase-4 (DPP-4) is a key enzyme involved in the regulation of glucose metabolism, making it an attractive target for the treatment of type 2 diabetes.

In this, we present a novel approach to design inhibitors using machine learning techniques to expedite the discovery of potent DPP4 inhibitors.

Utilizing a diverse dataset of known DPP4 inhibitors, we employed advanced machine learning algorithms to build predictive models.

These models were trained on molecular descriptors, physicochemical properties, and structural features of the compounds to discern patterns and relationships associated with inhibitory activity against DPP4.

The selected machine learning models demonstrated high predictive accuracy and robustness, as validated through cross-validation and external testing.

.

TEAM MEMBER DETAILS

<Student 1. 16893/B.Gayathri>
<Student 1. 7842343977>
<Student 1. vtu16893@veltech.edu.in>

INTRODUCTION

fTraditional drug discovery processes for developing DPP4 inhibitors are often time-consuming and resource-intensive. The integration of machine learning techniques in the drug discovery pipeline has shown promise in expediting the identification of novel drug candidates.

The integration of diverse datasets, including known DPP4 inhibitors, provides a foundation for training machine learning models. By leveraging molecular descriptors, physicochemical properties, and structural features, these models can discern subtle relationships and patterns associated with DPP4 inhibitory activity.The predictive accuracy and robustness of these models are rigorously validated through cross-validation and external testing. the successful training of machine learning models, the predictive power is harnessed to conduct virtual screening of chemical libraries.

The primary objective of this project is to employ a machine learning driven approach for the rational design of novel inhibitors targeting Dipeptidyl Peptidase-4 (DPP-4), with a focus on addressing the challenges associated with type 2 diabetes mellitus (T2DM). Spanning computational and experimental methodologies, aims to contribute not only to the development of novel DPP4 inhibitors but also to the broader field of drug discovery. By comparing the efficiency of the machine learning-driven approach with traditional methods, this project seeks to establish a benchmark for future endeavors, emphasizing the potential of data-driven strategies in drug development.

METHODOLOGIES

Data Collection and Preprocessing:

•Gather relevant data from various sources, including chemical databases, literature, and experimental assays.

Model Selection:

•Choose appropriate machine learning algorithms and models suited for the task, such as classification or regression models.

1.Cross-Validation:

1. Perform cross-validation to assess the generalization ability of the model and ensure it performs well on unseen data.
2. Use techniques like k-fold cross-validation or stratified cross-validation to split the dataset into multiple subsets for training and validation.

Iterative Development and Refinement:

•Iterate on the design and implementation of the machine learning approach based on feedback, experimental results, and domain knowledge.

RESULTS

The review finds that the structure of the dataset is relevant to the accuracy of the models, regardless of the selected features that are heterogeneous between studies.

Concerning the models, the optimal performance is for tree-type models. even tough they have the best accuracy, they require complementary techniques to balance data and reduce dimensionality by selecting the optimal features. Therefore,

K nearest neighborhoods, and Support vector machines are frequently preferred for
prediction.

Table 1. Label in 20pt Calibri.

Chart 1. Label in 20pt Calibri.

STANDARDS AND POLICIES

Anaconda prompt

This is a type of command line interface which explicitly deals with the ML(MachineLearning) modules.And navigator is available in all the Windows,Linux and MacOS.The anaconda prompt has many number of IDE's which make the coding easier. The UI can also be implemented in python.

It's like an open source web application that allows us to share and create the documents which contains the live code, equations, visualizations and narrative text. It can be used for data cleaning and transformation.

Algorithmic Governance:

Establish governance mechanisms for algorithmic decision-making processes, including model validation, monitoring, and version control. Implement accountability frameworks and review processes to ensure responsible use of AI technologies.

CONCLUSIONS

➤This project presents a pioneering approach to the design of novel inhibitors targeting Dipeptidyl Peptidase-4 (DPP-4) by leveraging machine learning techniques.

➤The literature review reveals the critical role of DPP-4 in the context of type 2 diabetes mellitus (T2DM) and underscores the limitations of traditional drug discovery methods. The integration of machine learning into the drug discovery pipeline offers a promising solution to expedite the identification of potent DPP-4 inhibitors

ACKNOWLEDGEMENT

Mr. V. Karthikeyan, M.E.,
Assistant Professor
Contact no: 9790256992

1. Project supervisor Mail ID

Figure 10.1: **Poster**

References

[1] Alaa, A. M., Bolton, T., Di Angelantonio, E., Rudd, J. H. F., van der Schaar, M. (2019). Cardiovascular disease risk prediction using automated machine learning: a prospective study of 423,604 UK Biobank participants. Nature Communications, 10(1), 1-13.

[2] Brnabic, A., Hess, L., Carter, G. C., Robinson, R., Araujo, A., Swindle, R. (2018). Methods used for the applicability of real-world data sources to individual patient decision making. Value in Health, 21(Suppl 1), S28-S34.

[3] Fröhlich, H., Balling, R., Beerenwinkel, N., Kohlbacher, O., Kumar, S., Lengauer, T., et al. (2018). From hype to reality: data science enabling personalized medicine. BMC Medicine, 16(1), 150.

[4] Grote, T., Berens, P. (2020). On the ethics of algorithmic decision-making in healthcare. Journal of Medical Ethics, 46(7), 462-467.

[5] Gawehn, E., Hiss, J. A., Schneider, G. (2016). Deep learning in drug discovery. Molecular Informatics, 35(1), 3-14.

[6] Hische, M., Luis-Domínguez, O., Pfeiffer, A. F., Schwarz, P. E., Selbig, J., Spranger, J. (2010). Decision trees as a simple-to-use and reliable tool to identify individuals with impaired glucose metabolism or type 2 diabetes mellitus. European Journal of Endocrinology, 162(4), 743-753.

[7] Hill, N. R., Ayoubkhani, D., McEwan, P., Sugrue, D. M., Farooqui, U., Lister, S., et al. (2019). Predicting atrial fibrillation in primary care using machine learning. PLoS ONE, 14(8), e0224582.

[8] Kim, I., Choi, H. J., Ryu, J. M., Lee, S. K., Yu, J. H., Kim, S. W., et al. (2019). A predictive model for high/low risk group according to Oncotype DX recurrence score using machine learning

[9] Luo, W., Phung, D., Tran, T., Gupta, S., Rana, S., Karmakar, C., et al. (2016). Guidelines for developing and reporting machine learning predictive models in biomedical research: a multidisciplinary view. Journal of Medical Internet Research, 18(12), e323.

[10] Vamathevan, J., Clark, D., Czodrowski, P., Dunham, I., Ferran, E., Lee, G., et al. (2019). Applications of machine learning in drug discovery and development. Nature Reviews Drug Discovery, 18(6), 463-477